Abstract

The purpose of this project is to explore some of the projectile problems behind rugby with particular reference to the Percy Grainger problem.

In rugby, kicking the ball has the big benefit of being able to gain field position by kicking the ball over the heads to their opponents and take play to where they want. However, by doing so, you usually give possession of the ball away, and with teams now filling themselves with players who have only ever known the modern professional game, keeping the ball is becoming harder and harder throughout a match.

It is therefore important that teams make full use of kicking the ball downfield and keeping it, or if it does go out of play, keeping the ball in the lineout situation. Kicking the ball downfield so your team is able to still challenge for the ball is a skill very useful in a game where possession count massively. Put that alongside the set lineout, this allows teams to systematically move their team downfield and into a scoring position.

This looks at the basic projectile angles required to allow players to know which angle kick/throw they should be doing to fulfil their need. It uses a computer algebra system called Maple, which has a model designed to help the lineout throwers find their teammates without the risk of the ball being stolen by the opposition. It also allows us to programme in different effects such as wind.

A model for a lineout throw has been developed for comparison with the Percy Grainger problem. A range of parameters have been tested to allow the derivation of feasible throw projectile angles to allow successful catching of the ball without it getting into opposition hands. For a typical lineout parameter, a projectile angle of between 50° and 60° was found to be appropriate. Despite its assumption and simplicity of the model, it could be used by players and coaches to explore lineout strategic tests.

Acknowledgements

I would like to thank my supervisor Dr Michael McCabe for his guidance and help throughout the time worked on my project. Michael helped me constantly to write my model, which enabled me to complete my project and form my conclusions. I would also like to thank Colin White, the author of *Projectile Dynamics in Sport* as he gave me ideas and feedback on my project which was very useful. Finally, the support of my friends and family has been amazing, as they knew I'd been struggling with this project, so to get the support I did was fantastic.

Contents

Chapter 1 Introduction to Rugby Projectile Problems

Ever since I heard the story of Percy Grainger, I have been fascinated with what it means and wondered if it is possible to adapt the fundamental projectile theory from this problem to the game of rugby. Certain play types like the Up-and-Under or more commonly known as a 'GaryOwen' kick uses the same principle of having enough hang time to be able to kick the ball high enough and far enough so the player can run down field and can catch it on its descent. Also, the Line-Out in rugby has a similarity as the thrower of the lineout has to be able to throw over the opposition jumpers/blockers to get the ball to their team-mate. This has already been explored using test subjects in (Trewarthaa, Casanovab, & Wilsonc, 2008) by looking at accuracy, velocity and joint movement. These projectile problems I hope to look at with a Percy Grainger look by finding angles of opportunity, but before this, I will explain some of the rules of rugby to show the value of why this project is being done.

Once this is explained, I will show you the working of the Percy Grainger problem, following and extending on what (White, 2011) and (Hart & Croft, 1988) did and solve it using a computer algebra system called Maple with different house hieghts. I will use the Percy Grainger model as a basis to help find an alternative model that will fit the required specifications of the projectile problems in the rugby lineout. These will also be solved in Maple, and will hopefully lead to a discussion about the results and models involved, along with the limitations. Finally, a conclusion will involve how good these models are relative to the real world, and how they could be improved to help replicate the conditions to become more helpful.

Rugby is a game played with two teams each having fifteen players, over 80 minutes on a pitch with dimensions in figure 1.1 below. The objective of the game is to score more points than the opposition by either scoring a try, which is grounding the ball on or over the try line, or kicking a conversion, penalty and drop goal over the crossbar between the posts. The main rule in rugby is that you can only pass the ball sideways or backwards, which means running forward with the ball is the easiest way to move up the field and score a try. However defences are so good in the professional era, which makes kicking the ball down field very important to gain good field position. This is why I have chosen to focus this project on the line-out and 'GaryOwen' kick because they are both key pieces of play that will help teams gain field position on a rugby pitch whilst keeping the ball.

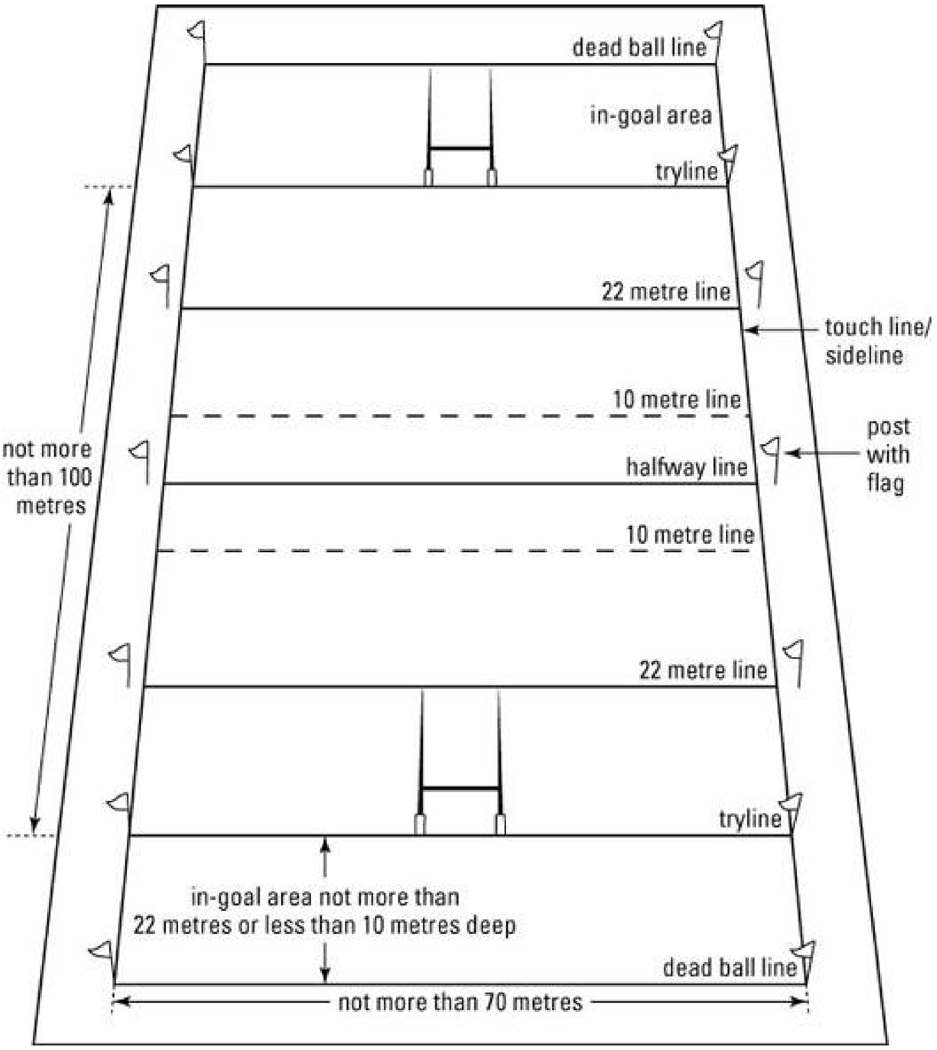

Figure 1.1: Dimensions on a Rugby Pitch

Source: http://www.athleticfieldmarker.com/rugbyfield#RugbyFieldCalculator

Usually when the ball is kicked, the team kicking the ball will lose possession as they are kicking the ball towards the opposition, which is why kickers tend to kick for touch to stop the opposition counter attacking. With players purposely kicking for touch, line-outs occur quite frequently during a match. If a team was able to dominate the line-out, by always keeping the ball on their throw in, and stealing the ball on their opponents throw, it will enable them to keep field position. However, sometimes players are not allowed to kick the ball straight out, so players have to keep the ball in play, which is where the 'GaryOwen' kick becomes useful. This style of play allows players to kick the ball down the pitch, whilst being able to run after it and challenge for the ball on its descent.

Chapter 2 Percy Grainger Problem and Solution

Percy Grainger was a well-known Australian composer born in 1882, Victoria, Australia. He was a renowned pianist, with popular works ranging from folk music in England, America and Denmark, to later composing some of the first kind of electronic 'synthesized' music. However, Percy also had a second passion after music, running. He loved the sport, but really he was a recreational runner, which is how he received the name of the jogging pianist. His interest in sport is where we look, as there is a story of him being able to throw a cricket ball over the roof of a house. Whilst the ball travelled over the house, he would run through the house and catch it on its way down (White, 2011). The problem is to find out whether this can be achieved humanly, in a drag less case.

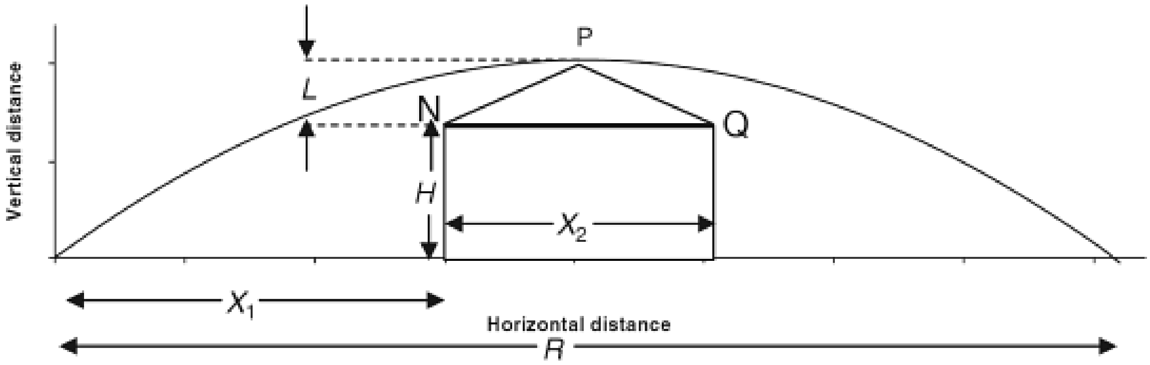

Figure 2.1: The Percy Grainger Problem

Source: (White, 2011)

We will start this problem by first deriving the formula for the trajectory of the projectile.

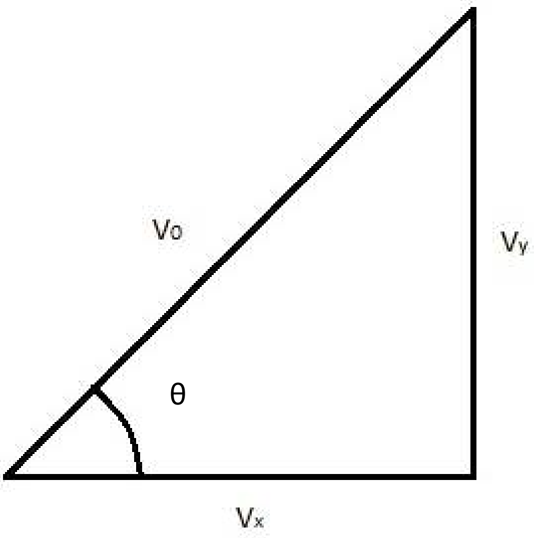

Figure 2.2: Initial Velocity given from its Vertical and Horizontal Components

From figure 2.2, we can find and in terms of by using basic trigonometry:

Equation 2.1:

Equation 2.2:

We now need to find out the distance a projectile will travel in both the horizontal and vertical directions given any time, t. Horizontally, this is using the simple well known equation , but vertically we have to take into account gravitational force, g. from this, we get equations:

Equation 2.3:

Equation 2.4: –

If we sub in our values for and the equation will then look like this:

Equation 2.5:

Equation 2.6: –

We can rearrange to give us a value for:

Equation 2.7:

By doing this we can combine the two equations, giving us a trajectory for our projectile.

Equation 2.8: –

This can then be cancelled down and simplified by using some trigonometry identities.

–

Equation 2.9:

With equation2.9, we can put it into the software programme Maple, along with typical conditions that would have been expected in 1882 to see if Percy Grainger could in fact throw a cricket ball over the roof of his house and catch it on the other side. Using notations in Figure 2.1, we change the equation to look like this:

Equation 2.10:

Where the distance has a number of constraints, so this is when

i) (point N)

ii) (point P)

iii) (point Q)

Where

- distance to the front of the house
- the distance to the back of the house
- launch velocity of the cricket ball
- height of the house
- height of the roof on top of the house
- acceleration due to gravity

are our initital conditions. Using Maple, we applied condition (ii) to equation 2.10, and came out with a range of angles that would clear point P. The range was . We then plotted the results so we could visulise the problem better, creating this graph.

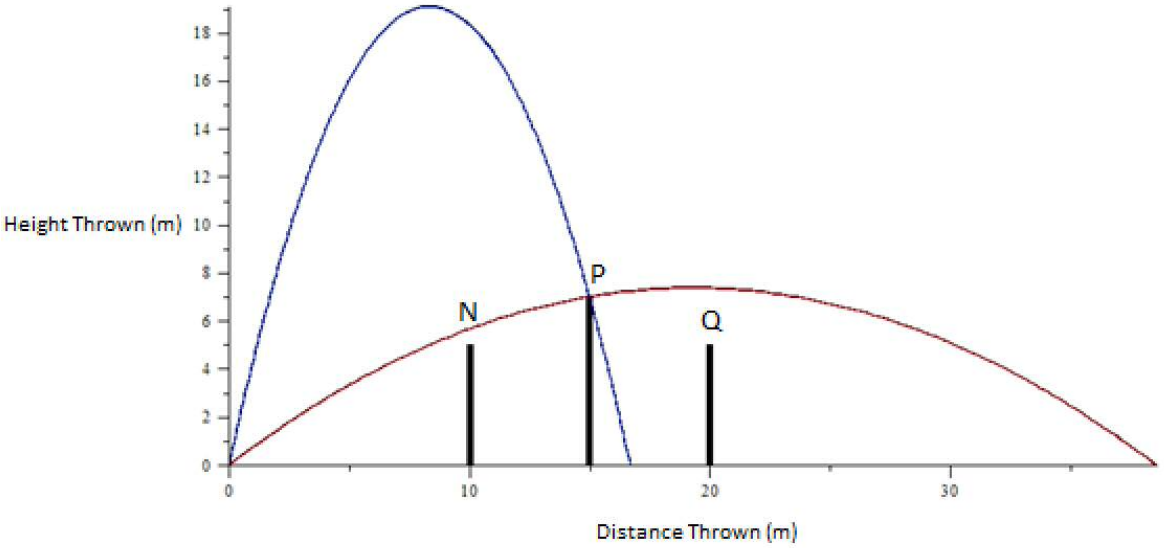

Figure 2.3: Graph showing condition 2 applied to Percy Grainger

As you can see, both angles clear point P just, along with point N, but the maximum angle falls short of point Q, meaning the angles we have now satisfy conditions (i) and (ii), but not (iii). Therefore, we applied condition (iii) to equation 2.10 and found the range for this condition, which was However, the minimum angle required for the ball to pass over point P was 37.36°, therefore the new range for the ball to be thrown so it clears all the points in figure 2.3 is With this new range, we were able to say the angle of opportunity for the Percy Grainger is possible is °.

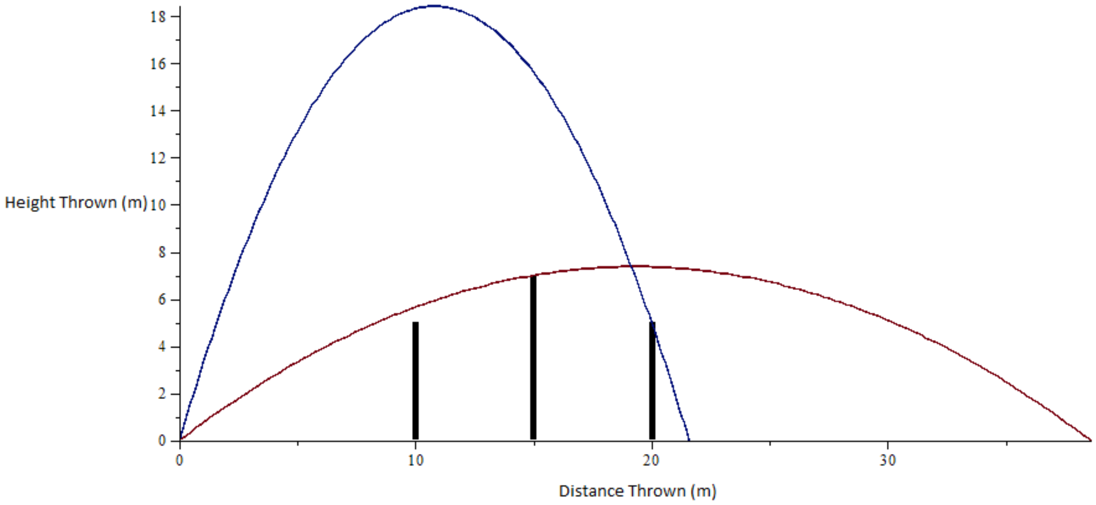

Figure 2.4: Graph showing final range to clear all points from Figure 2

From this graph, you can see how the ball clears all the points, but for the problem to be achievable, Percy Grainger had to be able to catch the ball on its decent. Now the possible angles have been found, we need to find the speed needed to run a distance that would allow him to catch the ball. Applying equation 2.1 to the extreme angles, this gives us the simple equation to find the horizontal velocities necessary for Percy to complete his challenge.

The average speed required for the minimum angle is far too large. Considering 100m Olympic sprinters run averagely at for a 10 second time, running is not realistic. However, running at is definitely achievable, so making sure he throws the ball at 73.68° he should be able to catch the ball the other side of his house.

If the house height was changed to 10 metres, the output of angles to find the angle of opportunity would be: . Therefore the angle of opportunity or success will be . This shows that increasing the size of the house by only 5 metres, double its original size, the angle of opportunity has nearly halved, so it is becoming more difficult to complete the Percy Grainger problem.

If the house height was changed to 20 metres, there are no angles available to clear the points; this is because there is not enough velocity to get the ball over this house. This is because the initial launch velocity is which means to clear a house to 20 metres high, we would need a much stronger initial velocity on the ball.

(Expansion on the steps shown in (Hart & Croft, 1988) and (White, 2011))

Chapter 3 Application of the Percy Grainger Problem to Rugby's Up-and-Under (GarryOwen) kick

Applying the Percy Grainger problem to rugby's 'GarryOwen' kick really makes the problem simpler, because a rugby pitch is flat with no physical obstructions like the house. When performing the kick in open play, opponents are the only obstacle. The tallest player would be around 2-2.1m tall, therefore finding the angle required to beat this is not a massive issue, but finding the maximum distance a player could gain whilst remaining competitive at regaining the ball is a useful because the point of the up-and-under kick is to retain possession. Therefore we would do this by finding the average speed of rugby players, more specifically the backs. We also need the maximum velocity a ball could be kicked, to find the hang time available. Finally it would be a matter of finding the distance a player could cover over that time, and then the angle required to kick the ball high enough so the ball lands where the player reaches. This chapter looks at the up-and-under 'GaryOwen' kick in a state of no drag, no wind and no spin.

Equation 3.1: $-$

Equation 3.2:

We set the y parameter to zero to find an equation for hang time:

$-$

$-$

or

Then to find the maximum range, we need to differentiate equation 3.1 with respect to t:

We then substitute this back into equation 3.2 to get:

Equation 3.3:

We have been given an initial velocity of the ball from (Lyall) and (TEACH P. , 2006), it has been found that punting the ball gives a change in velocity to the ball around - as the ball is dropped at around and the foot velocity is around . As we are looking to find the maximum distance achievable, we are going to take the initial velocity of the ball . Now, we will find the hang time of the ball by using our equation for hang time:

Now we need to know how far a rugby player can run in for this, we need to know how fast rugby players can run/sprint. I therefore did some research and found the RFU had done a study on the speed development of professional players, so I am going to use these numbers to have an average speed.

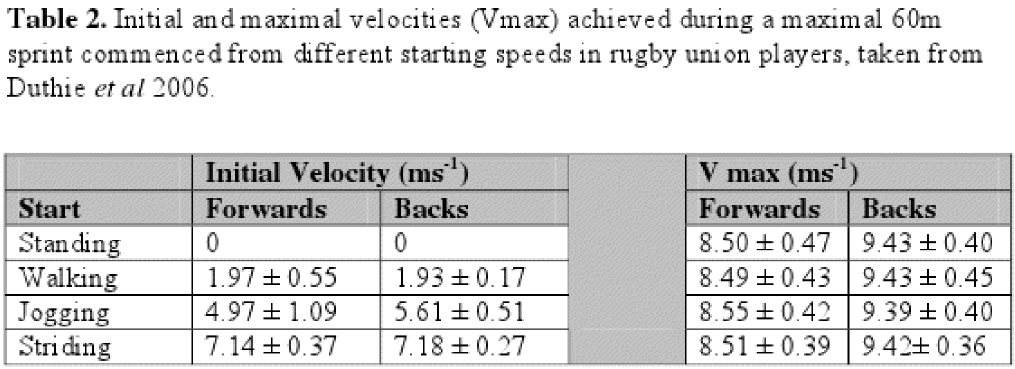
Table 2. Initial and maximal velocities (Vmax) achieved during a maximal 60m sprint commenced from different starting speeds in rugby union players, taken from Duthie *et al* 2006.

	Initial Velocity (ms^{-1})			V max (ms^{-1})	
Start	**Forwards**	**Backs**		**Forwards**	**Backs**
Standing	0	0		8.50 ± 0.47	9.43 ± 0.40
Walking	1.97 ± 0.55	1.93 ± 0.17		8.49 ± 0.43	9.43 ± 0.45
Jogging	4.97 ± 1.09	5.61 ± 0.51		8.55 ± 0.42	9.39 ± 0.40
Striding	7.14 ± 0.37	7.18 ± 0.27		8.51 ± 0.39	9.42± 0.36

Figure 3.1: Table showing players velocity over 60m

Source: (Pickering, 2007)

With this table, we are going to test the backs, as they are faster than the forwards, thereby they will be able to gain more ground over the given time. The average of the from the table, meaning our .

Therefore the ball must be kicked at an angle where , this will enable the backs to chase the kick and get under it to make a challenge and keep the ball. I shall also do it for the forwards, as they could also chase the ball just like the backs do, therefore

The ranges of these kicks are found by putting in these angles into our range equation 3.3:

When

When

As we can see, the ranges differ by about 10 metres, but are not very realistic as kicks of this type can go much further, and hang in the air for longer. From the results we have been given, the velocity we were given looks more like the velocity for a chip, as it goes in the air for a short period of time, and gains a fair bit of ground. This can be considered a very small ‘GaryOwen’ kick, not what it is usually referred to but still similar.

Chapter 4 Application of the Percy Grainger Problem to Rugby's Line-Out Problem

The line-out problem in rugby is similar to the Percy Grainger because jumpers can jump in between the thrower and his teammates. Therefore, this creates the obstacles like the Percy Grainger problem, however, this time we are not just trying to clear two or three obstacles, now we are trying to find our team mate above the launch height. This means we need trajectories of thrown balls to be able to clear blockers of any height while being able to find their teammate behind them. This needs a model, because the opposition can move in the lineout, so the first jumper could be tall at the front or the middle, you just don't know, so finding a model where you can input the values of the jumper so you can find out which angles the thrower has to make to clear them. We shall stick with the same conditions as the Percy Grainger problem, no drag and wind, apart from making the distances, velocity and height more relative to rugby. The launch height is being treated as zero, where in reality; the ball starts behind the head of the thrower, around 2 meters up. The jumpers are also able to reach the heights of 4/5 meters because in rugby, they are lifted up by their teammates, which is why they are able to get so far off the ground.

Therefore, we will continue to use equation 10 but our new initial conditions will be:

- , the distance to the first jumper.
- , the distance to the second jumper.
- , the distance to the third jumper.
- , the height of the lifted jumpers.
- , the extension of arms from the jumpers.
- , the launch velocity of the ball.
- , the acceleration due to gravity.

With sticking to the Percy Grainger framework, this would mean the tallest point in this lineout will be the middle; therefore, the inequalities will be:

- (Jumper 1).
- (Jumper 2).
- (Jumper 3).

From our inequalities, we can create this visualisation so we can have a clearer idea of what we want to achieve from our lineout.

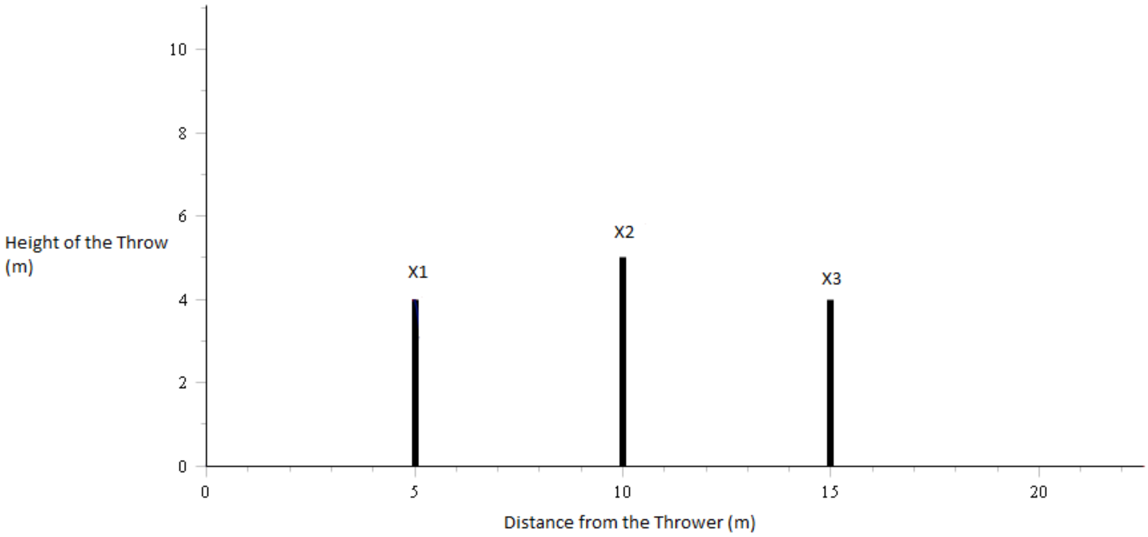

Figure 4.1: Layout of the Lineout Problem

From figure 4.1, you can see that the problem is very much similar in nature to the Percy Grainger problem above as it has the tallest obstacle in the middle, between two smaller points. However, a lineout can be put together in any order, so the tallest point in the lineout could either be at the front or the back, meaning the lineout problem could look like this:

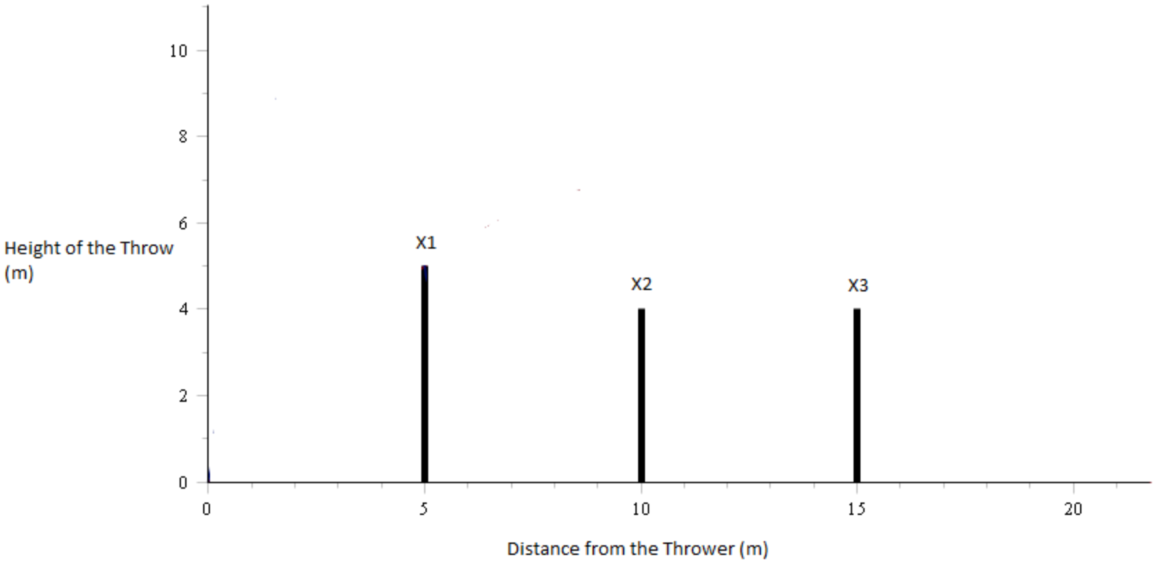

Figure 4.2: Layout of the Lineout Problem with being the tallest point

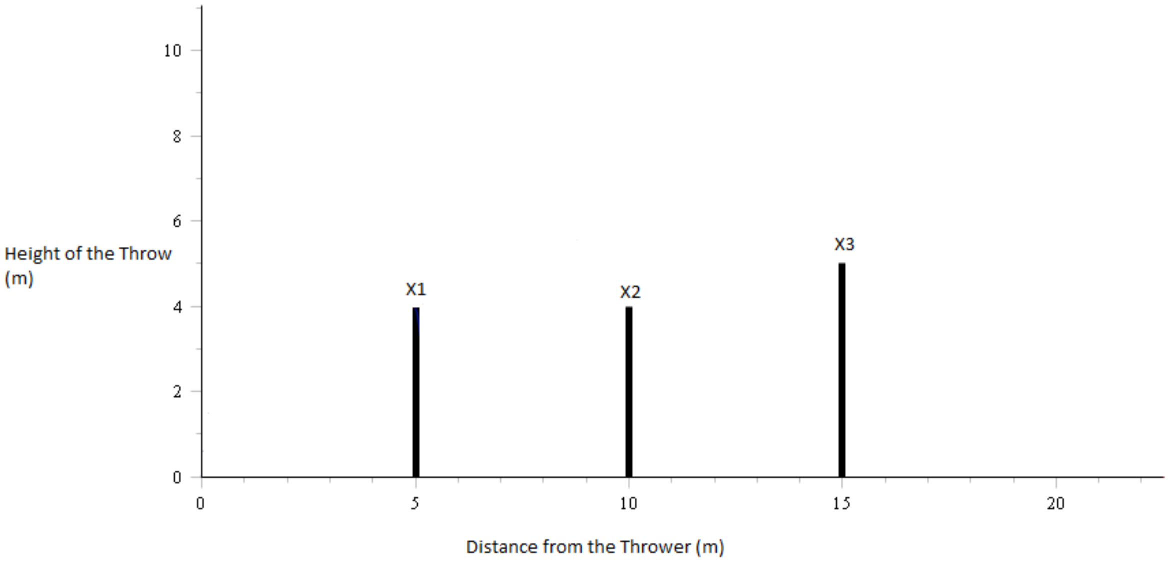

Figure 4.3: Layout of the Lineout Problem with being the tallest point

From figures 4.1, 4.2 and 4.3, the angles required by the thrower will differ depending on where the tallest jumper is. From the figures, we also have to change the inequalities, as the initial inequalities from this chapter are for figure 7, whereas the tallest jumper is in differing positions, the inequalities must change also.

Inequalities for figure 4.2:

- (Jumper 1).
- (Jumper 2).
- (Jumper 3).

Inequalities for figure 4.3:

- (Jumper 1).
- (Jumper 2).
- (Jumper 3).

From these new inequalities, we will be able to find out the angles to clear the jumpers, and find the angle required to get the ball to the teammate at the back. We are also able to use the Maple code written for the Percy Grainger problem, but with some small modifications so we find the angles required for the thrower. Below are the outputs from Maple, showing the angles from the throwers release.

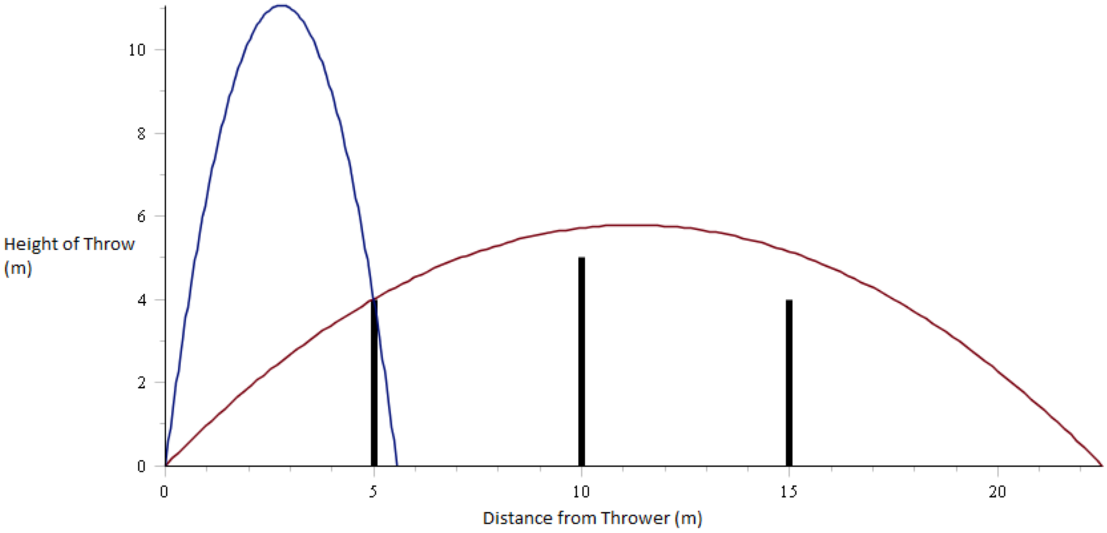

Figure 4.1.1: Angles clearing

The angle range for figure 4.1.1: . AOP = 37.04°

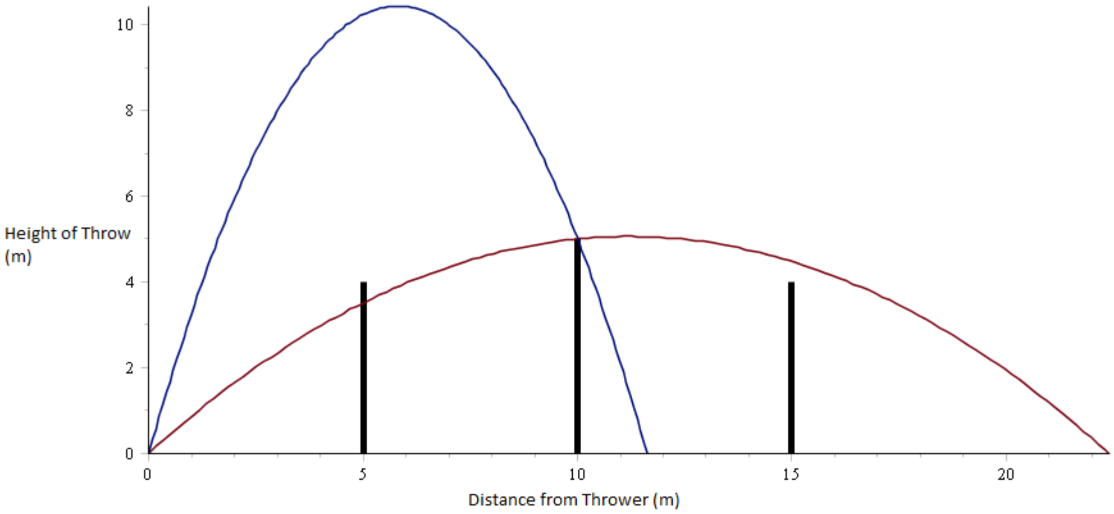

Figure 4.1.2: Angles clearing

The angle range for figure 4.1.2: . AOP = 32.36°

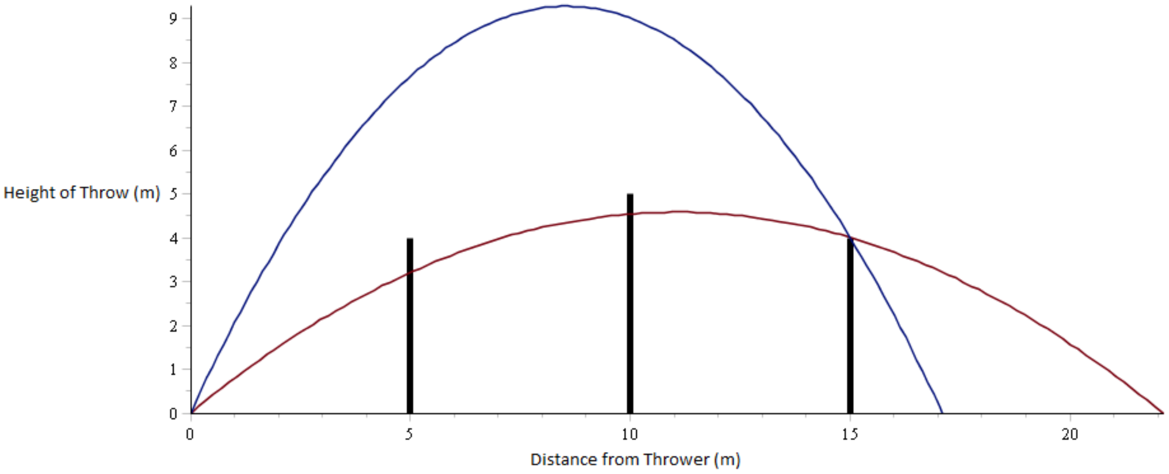

Figure 4.1.3: Angles clearing

The angle range for figure 4.1.3: . AOP = 25.61°

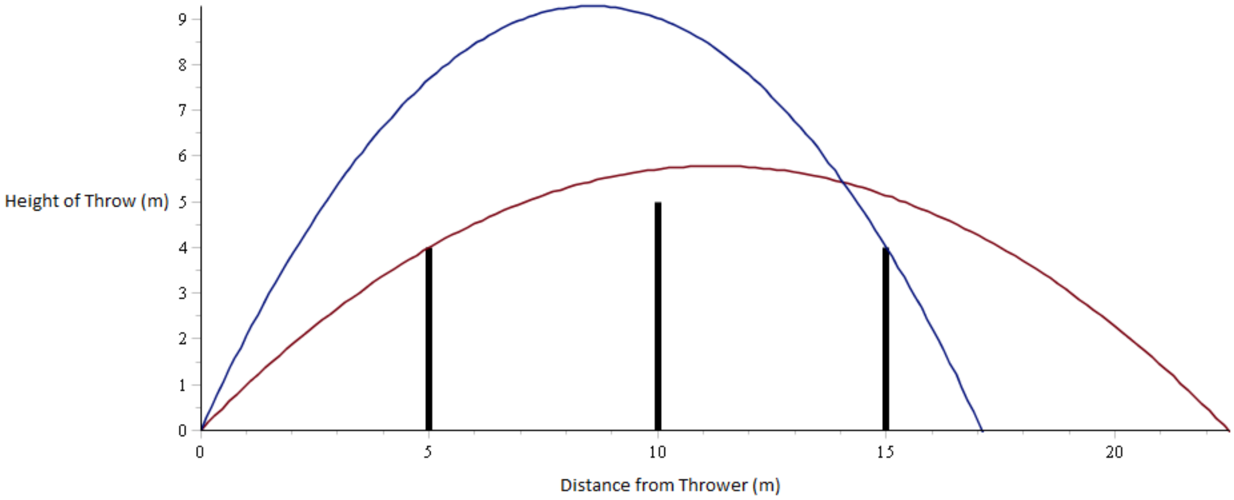

Figure 4.1.4: Angles clearing all Jumpers

The angle range for figure 4.1.4: . AOP = 19.46°

From figures 4.1.1, 4.1.2 and 4.1.3 above, you can see how they all clear their respective jumpers, but we needed a set of angles that cleared all three and that would go through the top of . We have this with figure 4.1.4, which shows the angles needed to throw completely over all three jumpers, along with the angle θ=65.27° as the angle required to throw over the opposition jumpers and into the hands of the teammate.

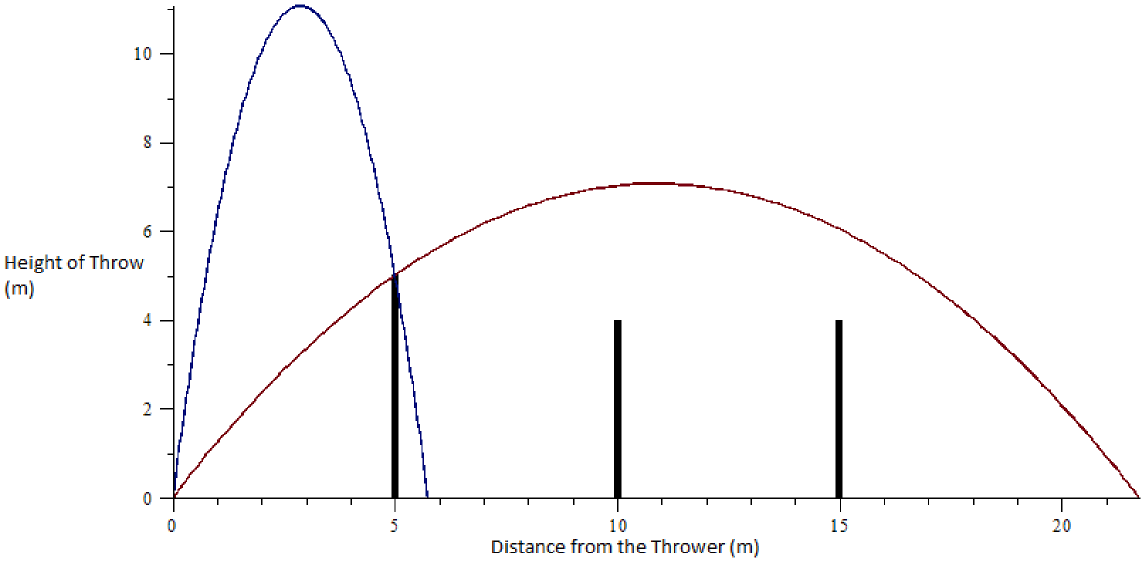

Figure 4.2.1: Angles clearing

The angle range for figure 4.2.1: . AOP = 30.21°

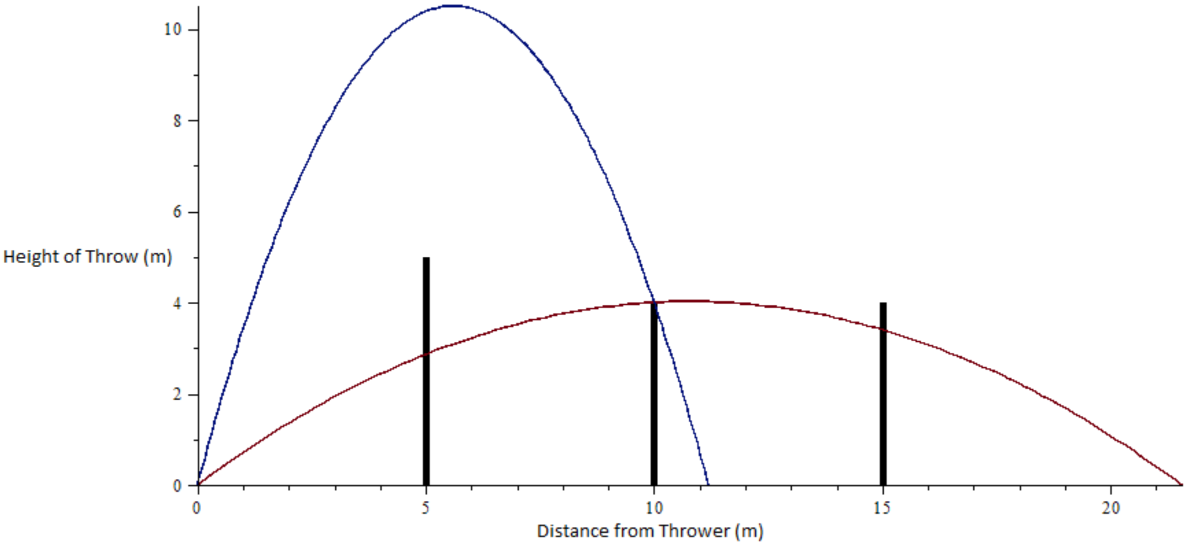

Figure 4.2.2: Angles clearing

The angle range for figure 4.2.2: . AOP = 38.36°

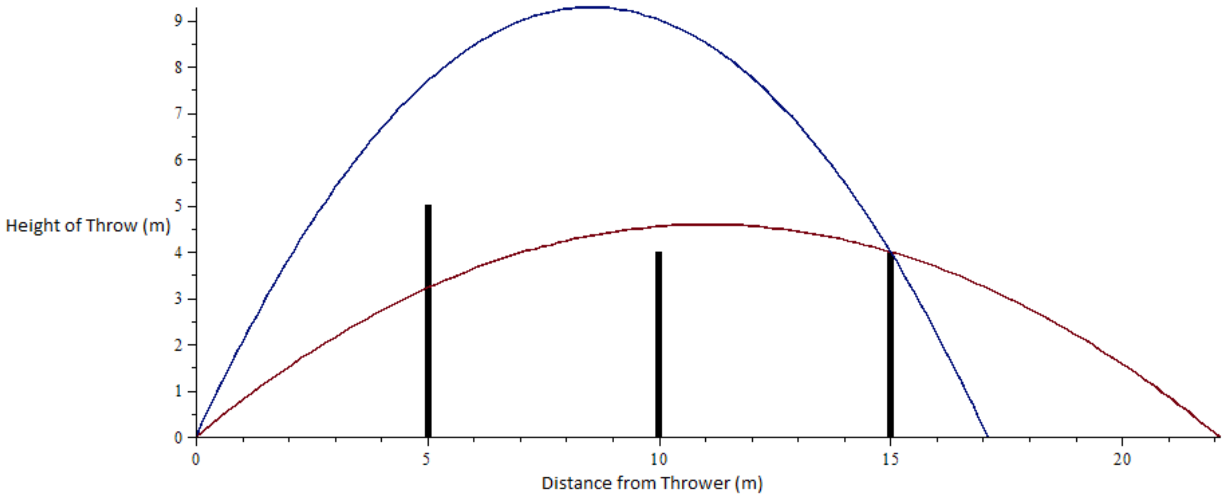

Figure 4.2.3: Angles clearing

The angle range for figure 4.2.3: . AOP = 29.61°

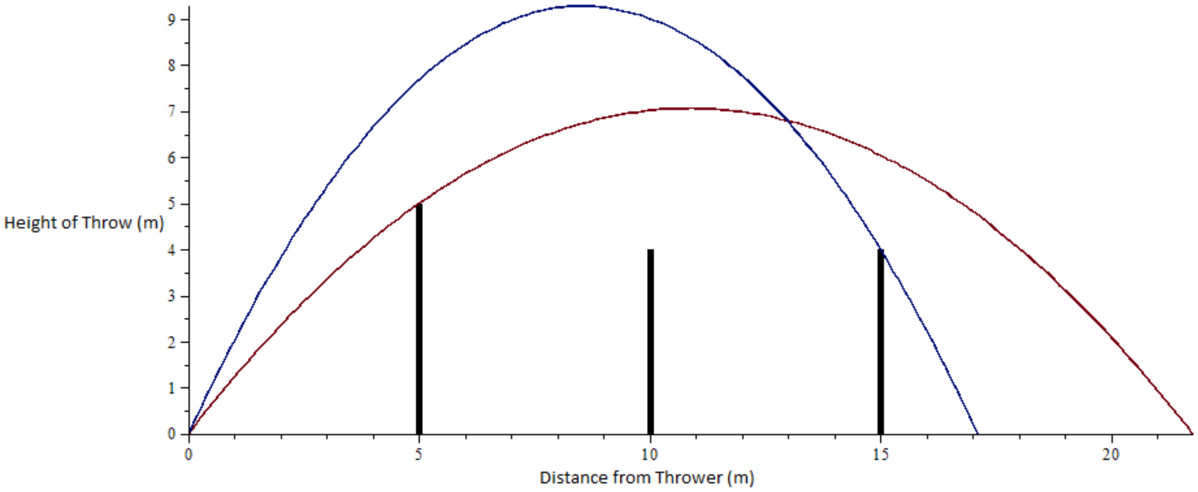

Figure 4.2.4: Angles clearing all Jumpers

The angle range for figure 4.2.4: . AOP = 12.87°

With the figures above, we can see how having the tallest player at the front makes the throw far more difficult. We can see only in figure 4.2.1 the ball clearing because it is specifically meant to, where neither 4.2.2 nor 4.2.3 does. However, figure 4.2.4 shows the angles needed to clear the first jumper, along with the same angle as figure 4.1.4, with .

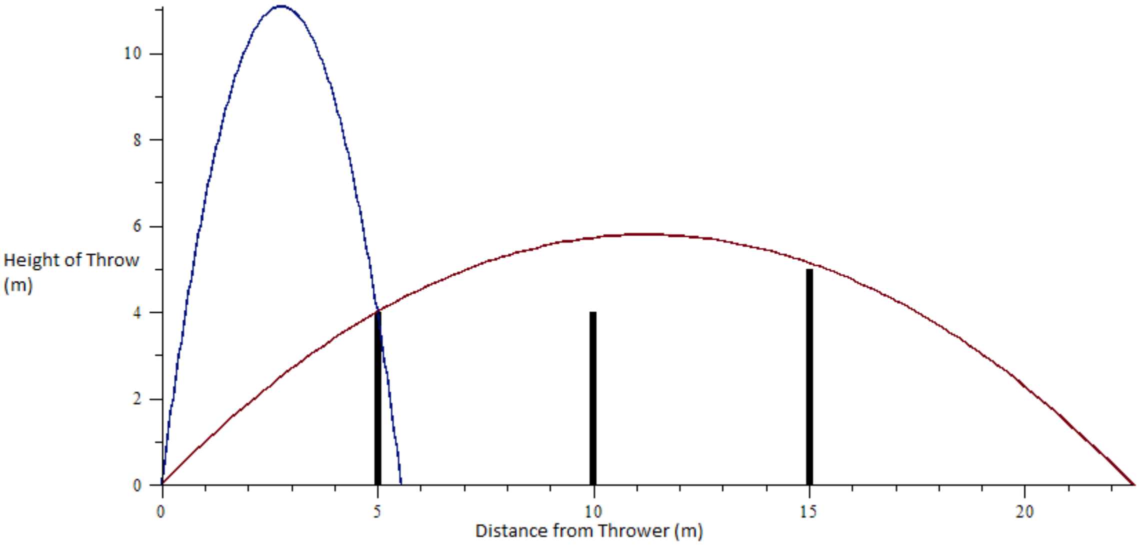

Figure 4.3.1: Angles clearing

The angle range for figure 4.3.1: . AOP = 37.04°

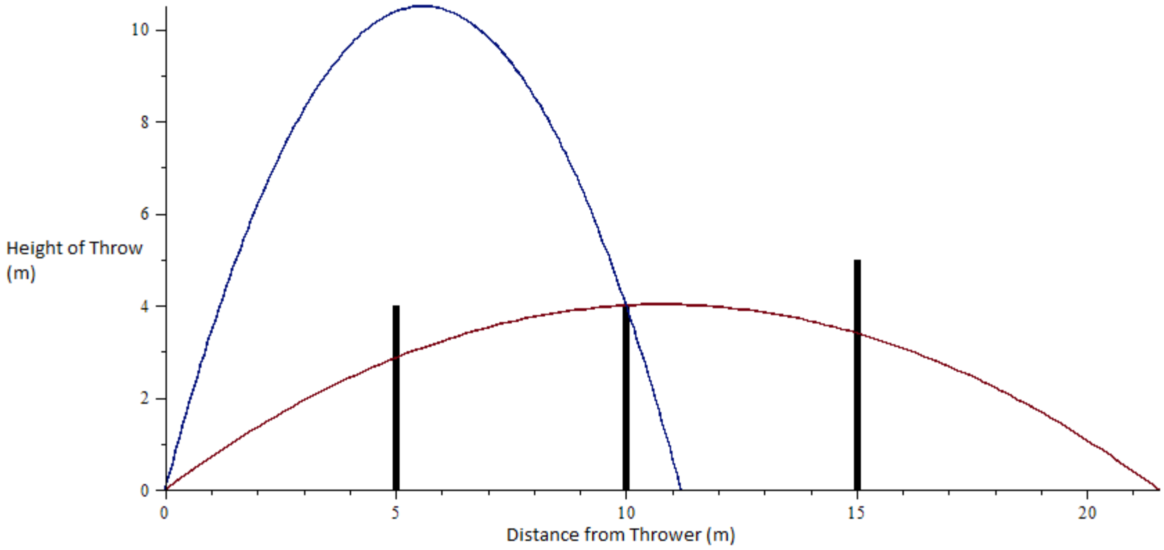

Figure 4.3.2: Angles clearing

The angle range for figure 4.3.2: . AOP = 38.36°

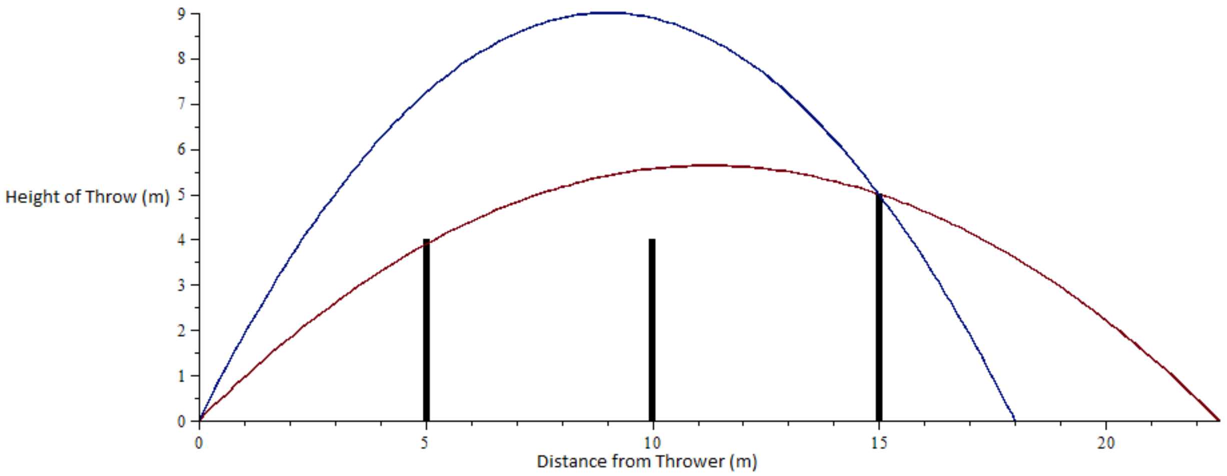

Figure 4.3.3: Angles clearing

The angle range for figure 4.3.3: . AOP = 18.43°

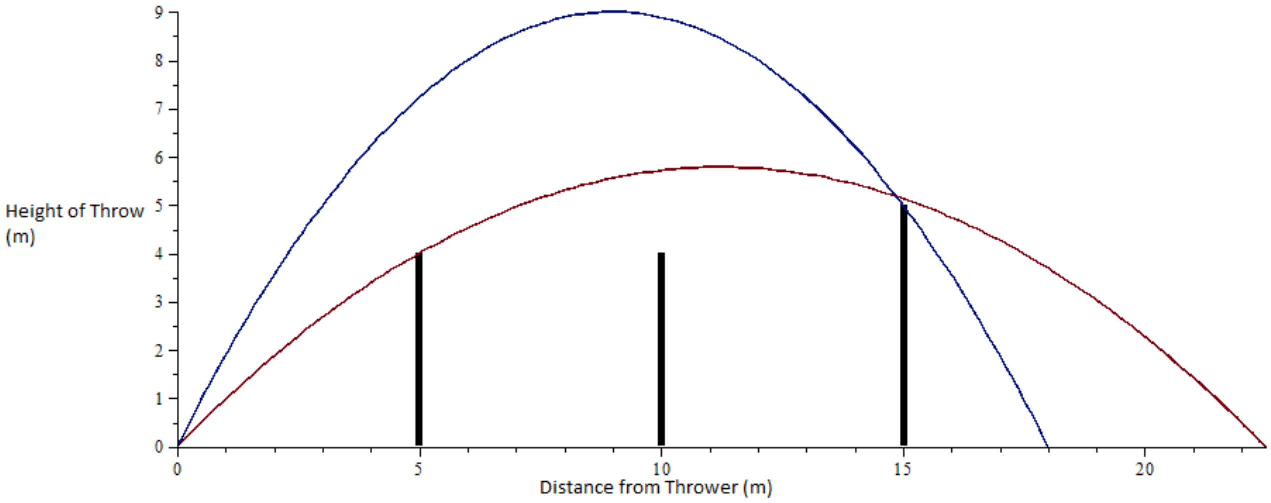

Figure 4.3.4: Angles clearing all Jumpers

The angle range for figure 4.3.4: . AOP = 17.62°

From the figures above we can see that it is by far easier having the tallest player on your team, as the angles nearly all clear the first jumper regardless. Again like the other figures above, figure 4.3.4 shows the angle of θ = 63.43° for getting the ball to our teammate.

Chapter 5 Rugby's Line-out Problem with Wind

Rugby is an outdoor winter sport, so it is very likely the conditions the players play in are windy, wet, cold or other traditional winter weathers. Therefore, it is unlikely we should treat the lineout in perfect conditions like we did in section 4. In this section, we will be changing the model to incorporate the factor of wind, as it is a bit more realistic to look at compared to wet or cold weather conditions as they will probably have more of an impact of the players well-being rather than the ball trajectory. Therefore we start like we did back in section 2, with the start-up equations:

Equation 2.1:

Equation 2.2:

Last time, we used these equations; we used them to enable us to find formulae for both the vertical and horizontal distances. These came out to be:

Equation 2.5:

Equation 2.6: $-$

However, because we are going to be taking into account wind, which is a horizontal variable, we need to edit equation 2.5 to take this into account. This means we get:

Equation 5.1:

Where W is the new wind factor and as it is speed, we add in another t variable as velocity is . This means that our equation for t will change, so our overall model will consequently change accordingly with the new factors. So rearranging our new equation we get a new equation for time:

Equation 5.2

As we now have equation 5.2, we do the same thing of subbing in our new equation into equation 2.6, to find the trajectory formula:

$-$

Equation 5.3

With equation 5.3, we will be able to put it into maple to enable us to solve the problem in the lineout, but now with slightly more realistic conditions. Therefore, as we are putting the ball up against a wind variable, we will change certain conditions, mainly the launch velocity, because if we throw at like in the previous example, against a headwind, the ball will not reach the player at the back. Therefore, our new conditions will be:

- , the distance to the first jumper.
- , the distance to the second jumper.
- , the distance to the third jumper.
- , the height of the lifted jumpers.
- , the extension of arms from the jumpers.
- , the launch velocity of the ball.
- , the acceleration due to gravity.
- , the speed of the headwind (- indicates headwind, positive would indicate tailwind).

From these initial conditions and our equation with the wind variable, we can find the angles now for trying to clear the opposition jumpers and get the ball to your teammate or over him. The inequalities we will be using will also be the same as the ones used in section 4, as they enable equation 5.3 to find the correct range.

Our first set of graphs will be with these inequalities, which is when the tallest jumper is at the front:

- (Jumper 1).
- (Jumper 2).
- (Jumper 3).

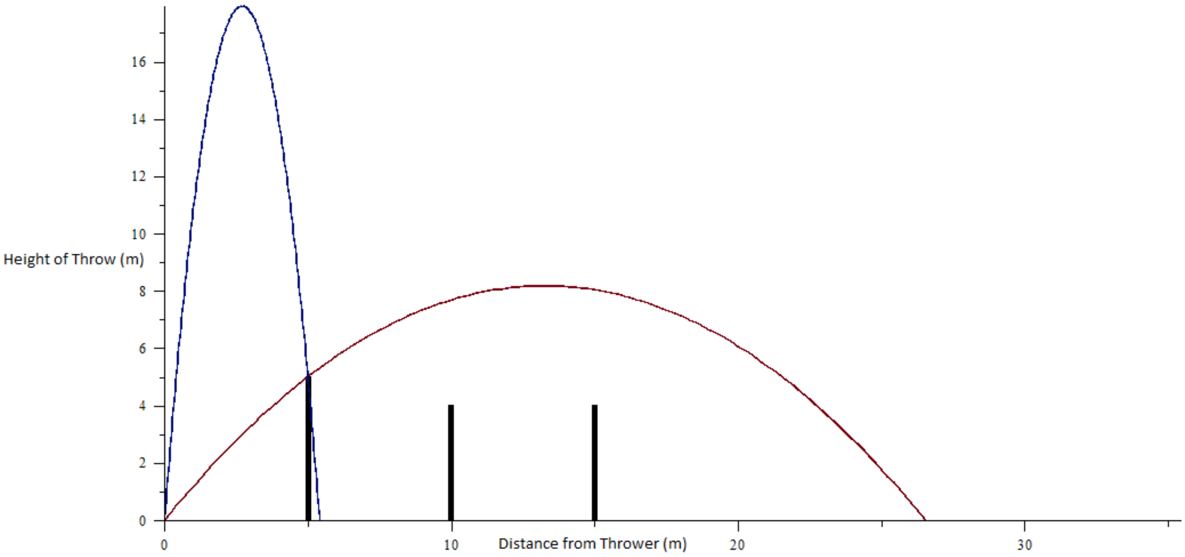

Figure 5.1: Angle range clearing

The angle range for figure 5.1 is: . AOP = 31.51°

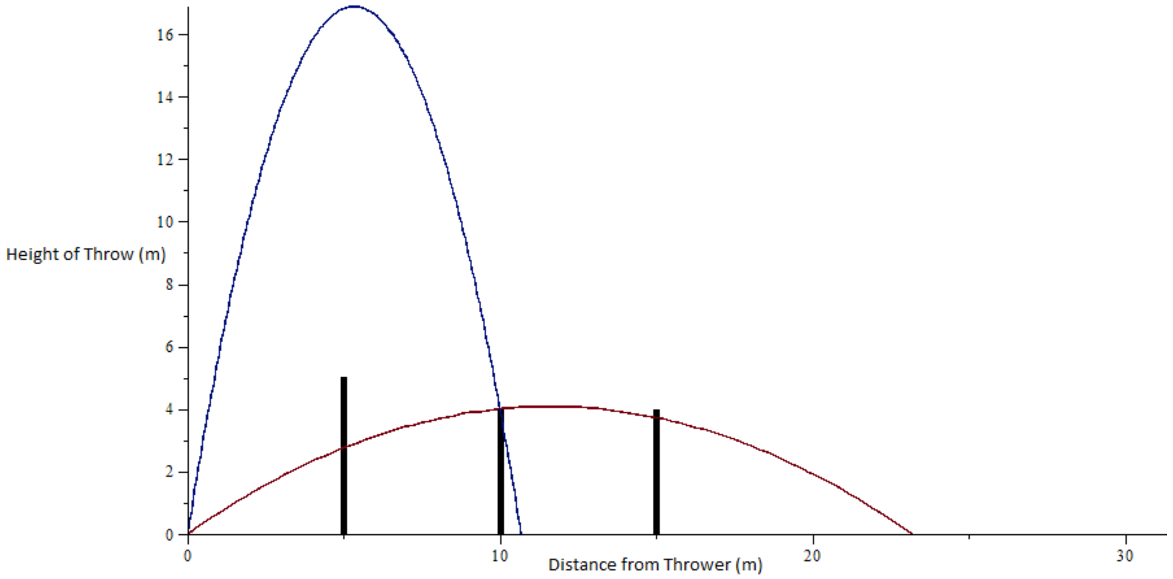

Figure 5.2: Angles clearing

The Angle range for figure 5.2 is: . AOP = 39.88°

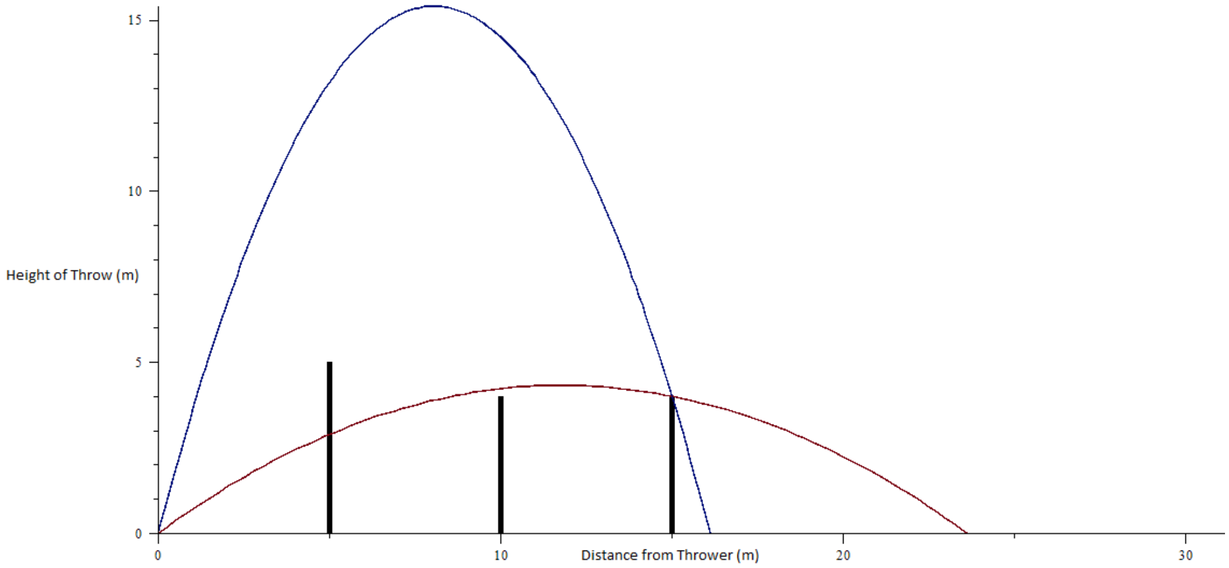

Figure 5.3: Angles clearing

The Angle range for figure 5.3 is: . AOP = 33.65°

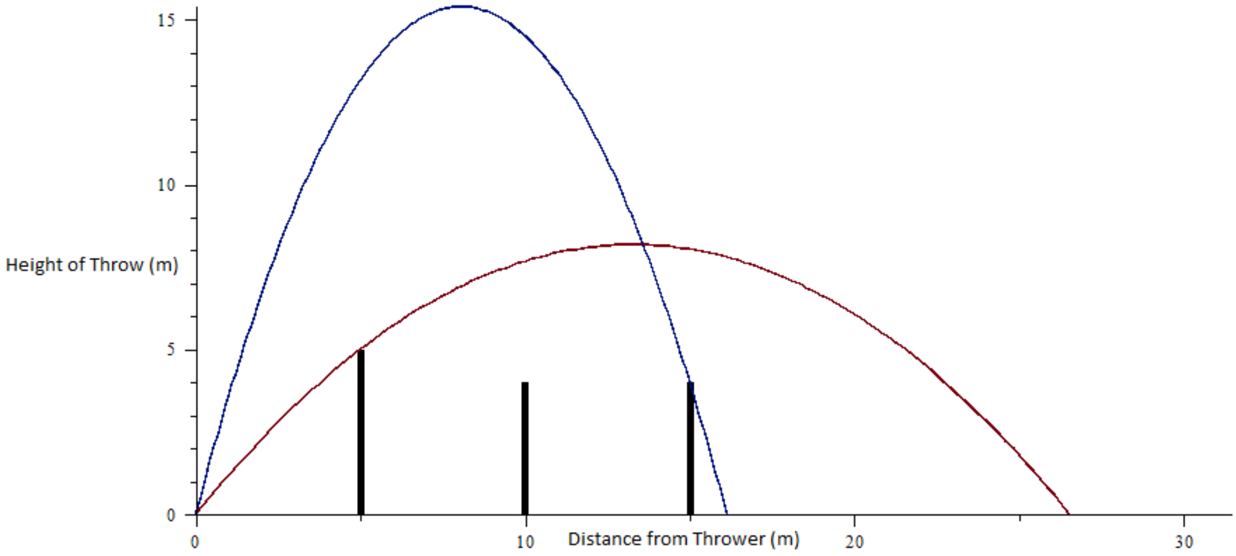

Figure 5.4: Angles clearing all Jumpers

The angle range for figure 5.4 is: . AOP = 21.59°

The following set of graphs will be with these inequalities, with the tallest player in the middle:

- (Jumper 1).
- (Jumper 2).
- (Jumper 3).

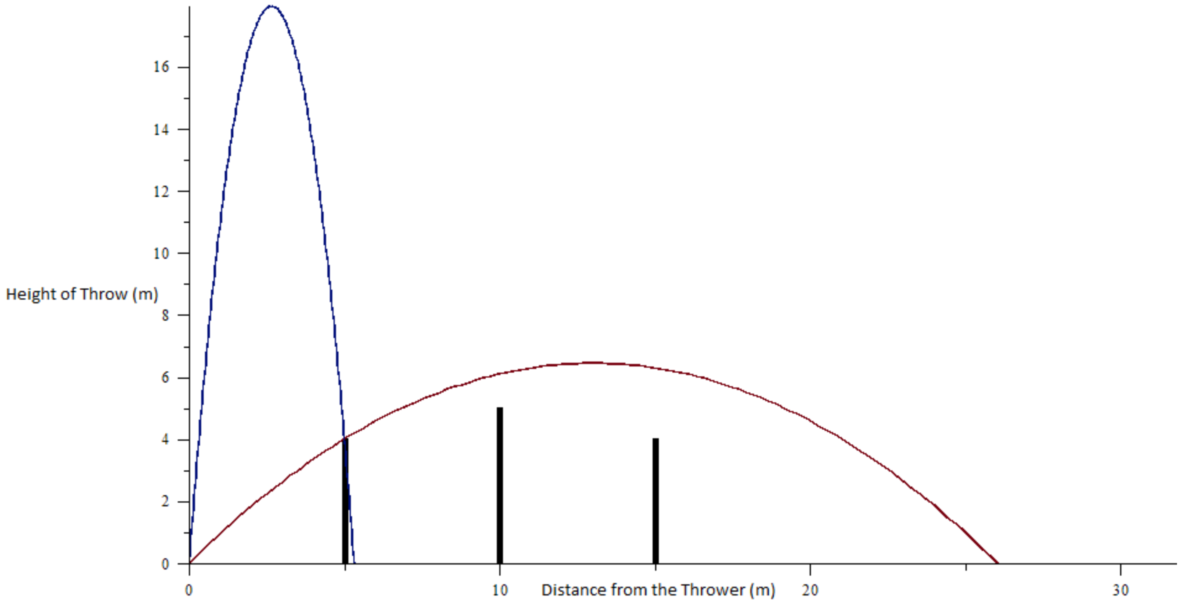

Figure 5.5: Angles clearing

The angle range for figure 5.5 is: . AOP = 36.74°

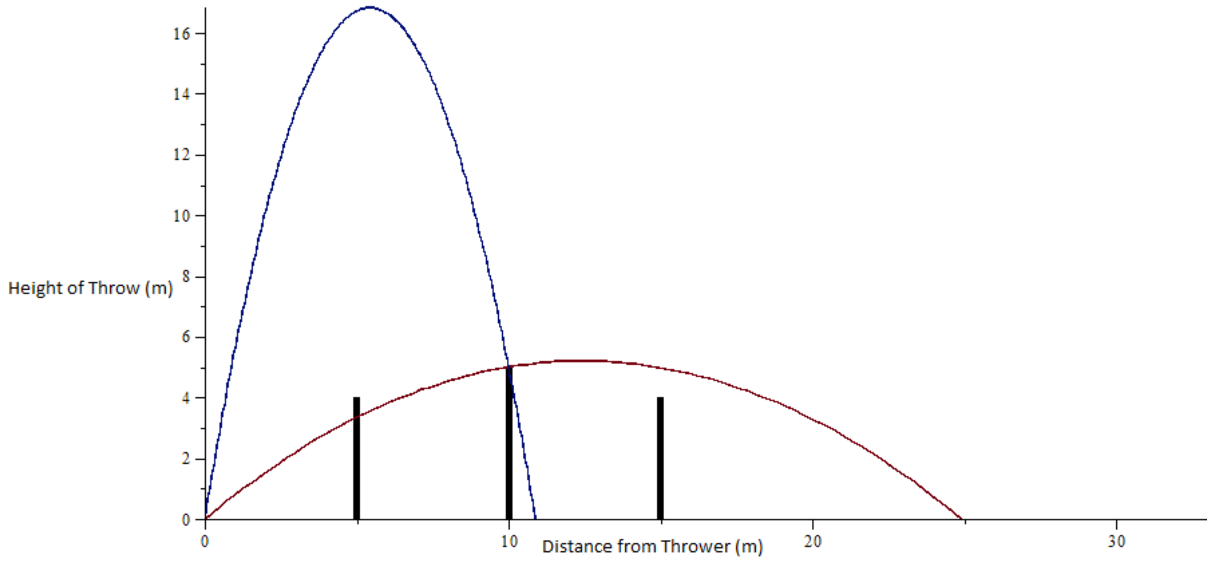

Figure 5.6: Angles clearing

The angle range for figure 5.6 is: . AOP = 35.88°

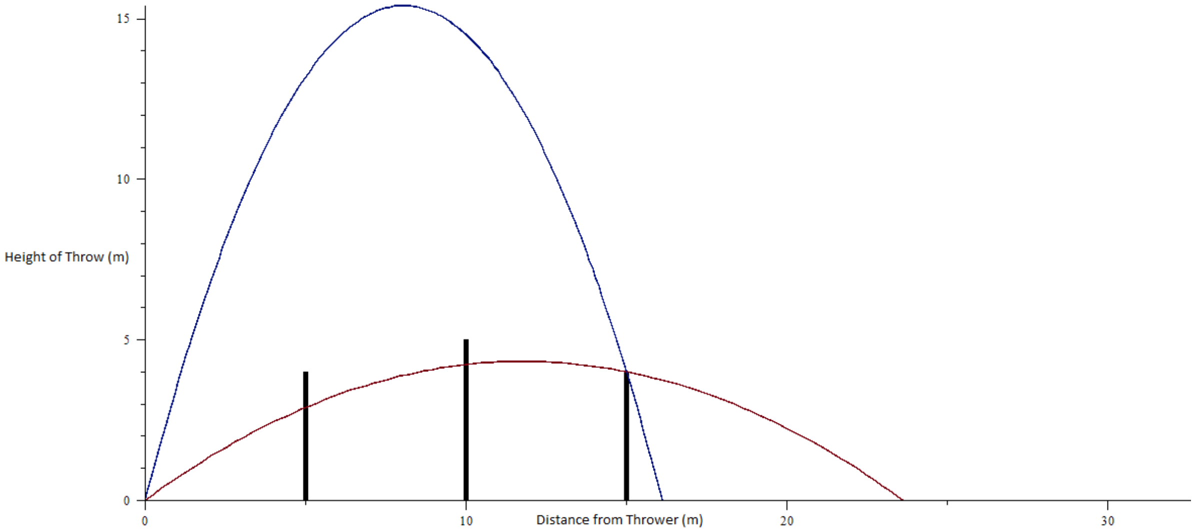

Figure 5.7: Angles clearing

The angle range for figure 5.7 is: . AOP = 33.65°

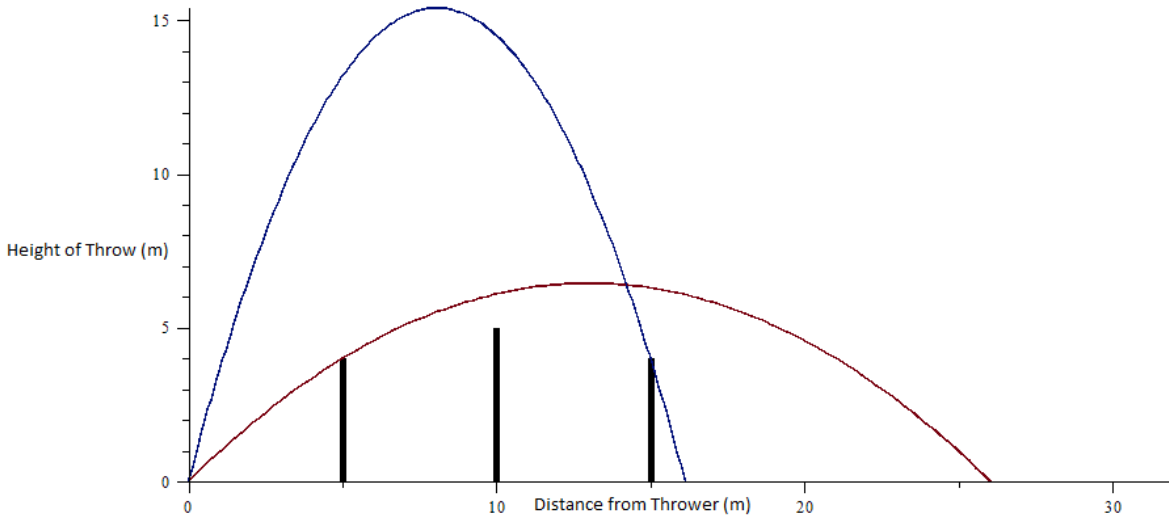

Figure 5.8: Angles clearing all Jumpers

The angle range for figure 5.8 is: . AOP = 26.74°

The following set of graphs will be from these inequalities, with the tallest player at the back:

- (Jumper 1).
- (Jumper 2).
- (Jumper 3).

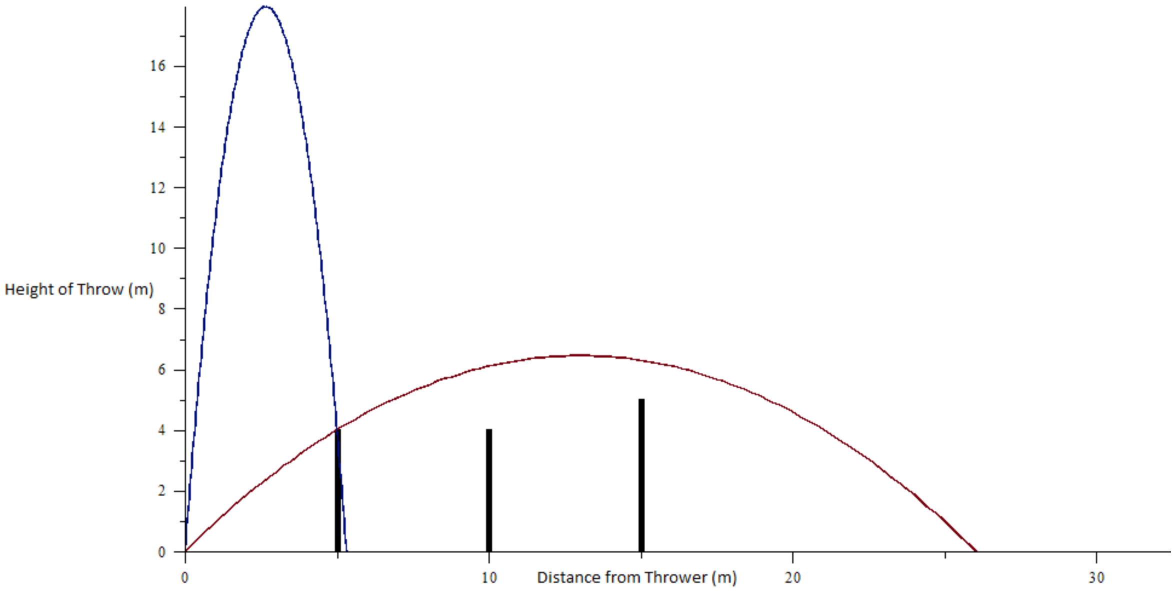

Figure 5.9: Angles clearing

The angle range for figure 5.9 is: . AOP = 36.74°

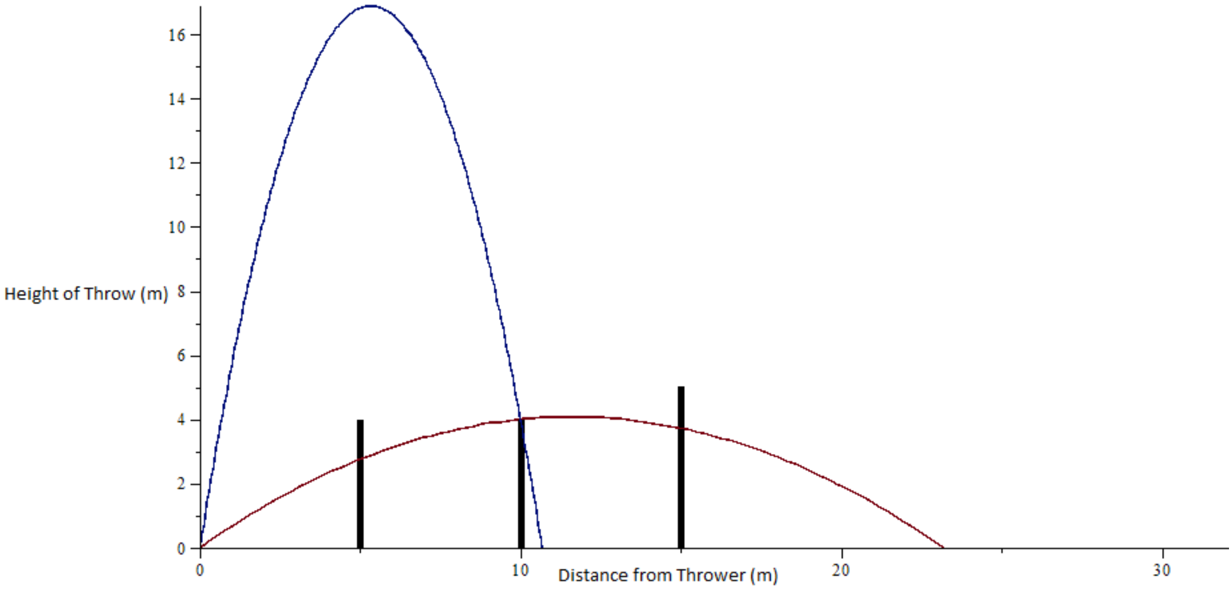

Figure 5.10: Angles clearing

The angle range for figure 5.10 is: . AOP = 39.88°

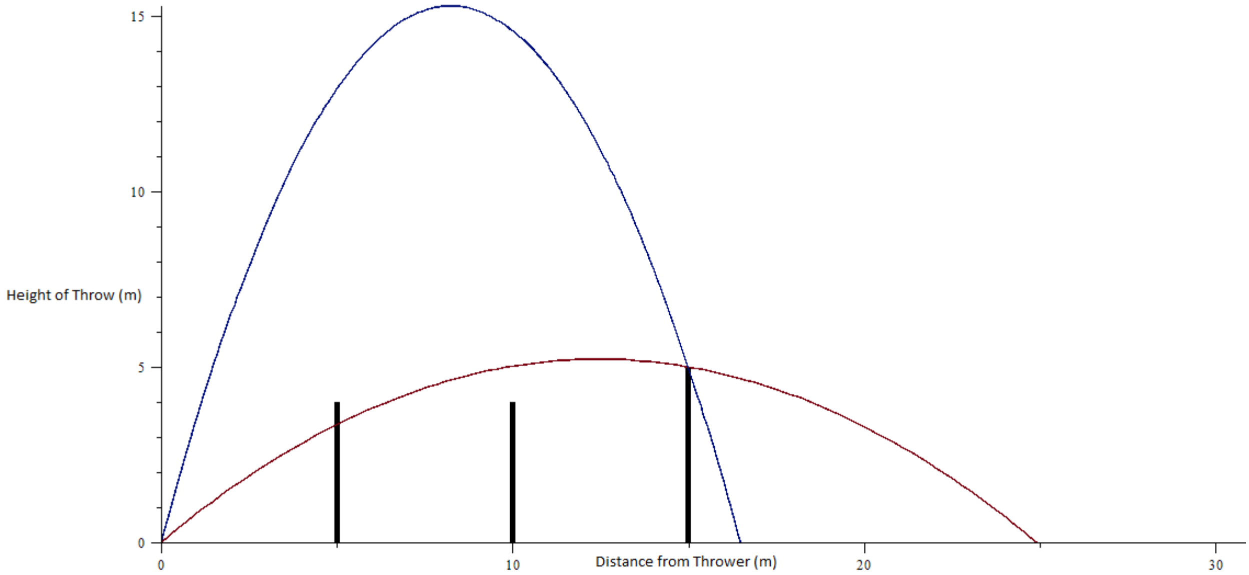

Figure 5.11: Angles clearing

The angle range for figure 5.11 is: AOP = 30.23°

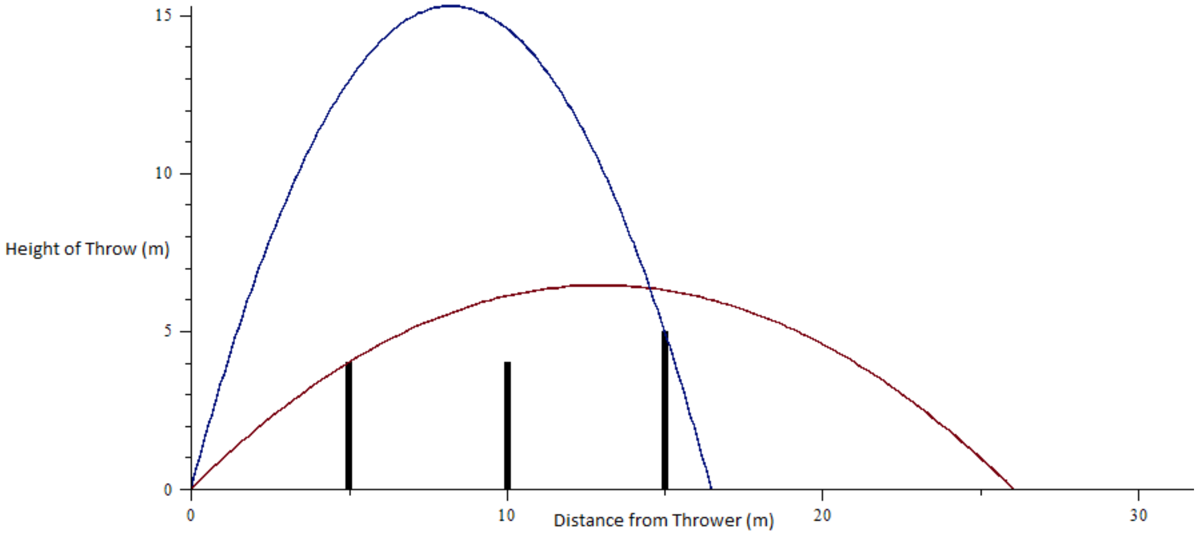

Figure 5.12: Angles clearing all Jumpers

The angle range for figure 5.12 is: . AOP = 26.35°

From the graphs in this section, we can see the angles required by the thrower to clear the opposition jumpers and find their teammate at the back of the lineout with the tall player in their different positions. We can also see that the curve itself is not a straight forward parabola, which shows that the wind variable has had an effect on its shape, making it more realistic.

This model has the ability to check the angles required for many different velocities and head/tail winds as the conditions can be easily changed. If you look at appendix 3, you will be able to see how the can be changed to enable coaches and players alike to find the necessary angles for the throw. With the two extreme angles of each graph, we have the 'flat' and 'lob' throw for each permutation, which will enable the thrower to keep the opposition guessing. The 'flat' throw will get the ball to the teammate more quickly, resulting in quicker attacking ball which is what the team wants. However there is the risk of it being intercepted by the opposition, so the safe option is to 'lob' the throw by using the maximum angle, however this has a bigger hang time so the attack will not be as effective.

Chapter 6 Conclusion

This project show the theoretical theory behind the Percy Grainger problem, showing that it is indeed possible for a person to throw a ball over one or more players and catch it the other side. However, there is a limit when the Percy Grainger problem is impossible, because the minimum angle is too small and would not allow enough time for Percy to run and catch it. This limit could either be a minimum angle like here, or a maximum height, such as throwing a ball with enough velocity that it will require more force the higher the physical object is you are trying to clear.

This project also gives a model for players and coaches to use to give them an initial idea of some of the angles the lineout thrower could utilize to avoid the opposition stealing the ball. It is very easy for them to manipulate any of the parameters to enable them to find out what kind of angle they will need. This can be used for players of all levels of experience and skill, so beginners have more idea of what they have to do, or experienced players' fine tuning their technique to try and make sure they utilize every lineout to its maximum potential.

Overall, this project can be used as a starting point for beginners, or for players with more experience hoping to get answers as to why they are not performing as well as they can. Therefore, basic training in this set play, will enable players of all experience and talent to get the perfect throw for each and every lineout.

Limitations

There are a few limitations through this project, which are explained below.

1) This project only looks at the hypothetical theory, and used examples of the model to show how it worked. It does not use the model to create data, or solve the problem to find a given set to angles that the opposition will never get as there are too many permutations to consider with just this simple model.

2) The lineout problem only looked at the fundamental front middle and back locations, however, the jumpers can be lifted anywhere between the 5m and 15m lines. This would require a programme to run a loop or sequence to enable the model to check the angles required for these different distances from the thrower. This would give a clearer picture of the angles available to the thrower, so they knew exactly where the opposition would not be able to steal the ball.

3) The lineout problem also only looked at one player being very tall/big, where teams seem to be finding taller and taller player to put in their lineouts. Therefore, there was no study on what kind of angles would be needed if the thrower was small and the all the jumpers were tall, but this can be looked at through the model given at the end of this project, it is just about having the time to look through all the permutations.

4) During this project, the velocity was kept constant during each section of this report, making it easier to fit, but difficult to judge as they were constants chosen by me from my experience of playing rugby. This didn't make for vary realistic angles for hitting teammates, as the hang time of the ball would be too long. Usually the ball is thrown much flatter, but for the simplicity of this project, we had to keep the jumpers/obstacle in place.

Chapter 7 Further Work

1) During this project, we did not look at the way the ball rotated and spun, which can have a major effect on how the ball will travel through the air. Whether the ball tumbles, going end over end, or torpedoes, spinning to keep its aerodynamic shape, the ball will travel differently with air resistance as one will cut through the air far better than the other.

2) Playing rugby can be an emotional and psychological experience, especially at professional and international level. Therefore, looking at how players perform the tasks under the pressure when one single mistake could cost their team the match would be an interesting factor to analyse.

3) Players do not play rugby for one play and get a rest like other sports, it is continuous and tiresome. Therefore, players will start to feel exhausted after running, tackling and getting tackled over the 80 minute match. Being able to perform these tasks refreshed is one thing, but after 60/70 minutes of hard work may be different. Cross examining results of fresh players to results of tired players would be interesting.

4) In the laws of rugby, players are not allowed to jump or be lifted until the ball has begun moving forward as a result of being thrown. Therefore, the players require timing to make sure they are at the height of their jump as the ball reaches them to enable them to catch the ball. This is a variable not explored, but would make for some interesting results if you examined success rates also.

5) One of the recent lineout journals I read (Trewarthaa, Casanovab, & Wilsonc, 2008), it looked at the professional game and noted how important the lineout being, saying that the team with possession retains possession 80% of the time, and that 26% of scores come from plays off winning their own lineout. It goes on to gain data on launch velocity, accuracy, throwing technique and percentage change in the joints when the test subjects throw the ball. It is interesting, and it is the next step onward from this kind of project, as it takes a look at real data rather than hypothetical theory.

Bibliography

De Mestre, N. (1991). *The Mathematics of Projectiles in Sport.* Cambridge: Cambridge University Press.

Hart, D., & Croft, T. (1988). *Modelling with Projectiles.* Chichester: Ellis Horwood Limited.

Joyce, D. (1997). *Summary of Trigonometric identities*. Retrieved February 1, 2013, from Clark University website: http://www.clarku.edu/~djoyce/trig/identities.html

Lipscombe, T. (2009). *Physics of Rugby.* Nottingham: Nottingham University Press.

Lyall, D. (n.d.). *Rugby.* Retrieved 1 30, 2013, from Information and Education for Coaches: http://www.coachesinfo.com/index.php?option=com_docman&task=cat_view&gid=45&Itemid=76

Moritz, E. F., & Haake, S. (2006). *The Engineering Of Sport 6* (Vol. 1). New York: Springer Science & Business Media.

Moritz, E. F., & Hakke, S. (2006). *The Engineering of Sport 6* (Vol. 2). New York: Springer Science & Business Media.

Pickering, S. (2007). *Speed Development in Rugby Union Players.* Rugby Football Union.

TEACH, P. (2006, May 4). *Kicking in Rugby.* Retrieved January 30, 2013, from Physics of Rugby: http://physicsrugby.blogspot.co.uk/2006/05/kicking-in-rugby-mathematical.html

TEACH, P. (2006, August 23). *Lineout.* Retrieved February 2, 2013, from Physics of Rugby: http://physicsrugby.blogspot.co.uk/2006/08/lineout.html

Trewarthaa, G., Casanovab, R., & Wilsonc, C. (2008). A Kinematic Analysis of Rugby Lineout Throwing. *Journal of Sports Sciences*, 845-854.

White, C. (2011). *Projectile Dynamics in Sport.* Oxon: Routledge.

Appendix 1

Final Year Project Plan (U02849)

Date: 7/11/2012

Student: Nick Weatherseed 487586

Supervisor: Colin White / Michael McCabe

Provisional Project Title:

Mathematical Projectiles: The effects of the ‘up-and-under’ rugby kick.

Project Brief:

This project is looking at the ‘up-and-under’ kick in rugby, which when kicked has a parabola shaped curve. I will be looking at this kick with the intentions to see what the optimum distance gained whilst kicking the ball up and catching it on its way down. Once a basic model has been found, I will then test the model by adding in variables to see if anything changes due to the effect of wind, as rugby is played in all kinds of conditions. I also hope to include drag, as a rugby ball in real life doesn’t travel in a perfect parabola; it goes straight and then drops quite quickly. So including drag on one axis should make the parabola look more like what the rugby ball actually does.

For project, I hope to include a history into why kicking in rugby has become far more technical and important than it used to be. I will also include an introduction of projectiles, so readers that don’t understand how the parabola curve is found will have some understanding. Another factor which will be taken into account is the human physical limitations, as ordinary people or professional rugby players have a limited speed and strength.

Finally, I would like to have a look to see if the Percy Grainger Feat is possible with the ‘up-and-under’ kick. For this, I will use my model calculated from the previous chapters and see if I can adapt it to create a collision model. I mainly want to see if it is humanly possible, as it has been calculated to be mathematically possible.

Plan:

9th November – Handing in Project Plan

23rd November – Draft of Introduction / Familiarise with LaTex

30th November – Introduction Complete

14th December – Draft of Preliminary Projectile Theory

18th January – Preliminary Projectile Theory Complete

1st February – Draft of Optimum Parabola

8th February – Optimum Parabola complete

1st March - Draft of Percy Grainger Feat

15th March - Percy Grainger Feat complete

22nd March – Conclusion / Bibliography / Abstract

28th March (Due Date) – Checks / Fix problems / Bind and submit

Background reading and Resources:

Armenti, A. Jr. (1997) The Physics of Sport. New York: American Institute of Physics

White, C. (2011) Projectile Dynamics of Sport: Principles and Applications. London: Routledge

Danby, J.M.A (1997) Computer Modelling: From Sports to Spaceflight – From order to Chaos. Richmond, Va: Willmann-Bell

Berry, J. Houston, K. (1995) Mathematical Modelling: Modular Mathematics.London: Edward Arnold

Haake, S. Moritz, E.F (2006) Engineering of Sport 6: Developments for Sports. Volume 1. New York: Springer

Haake, S. Moritz, E.F (2006) Engineering of Sport 6: Developments for Disciplines. Volume 2. New York: Springer

De Mestre, N. Australian Mathematical Society (1990) The Mathematics of Projectiles in Sport: Lecture Series 6. Cambridge: Cambridge University

Hart, D. Croft, T. (1988) Modelling with Projectiles. Chichester: Ellis Horwood

Project Report Chapters:

Abstract – short summary

- Motivation/problem statement
- Methods/procedure/approach
- Results/findings/product
- Conclusion/implications

Introduction

- Aims and Goals
- Brief history to rugby, kicking and its importance
- What programs I will use
- What each chapter will contain

- Any human constraints – running speed or kick velocity?
- Drag, how it changes and effects balls trajectory

Preliminary Projectiles Theory

- Introduction to the projectile parabola
- The enveloping parabola
- Velocity and position vectors
- Show how drag and the wind force can affect the parabola
- The pros and cons of the variables

Calculating the optimum parabola

- Parabola of safety/ bounding parabola
- Inclusion of drag/resistance in the parabola
- Which variables will be in use
- Analysis and reaction of the results

Is Percy Grainger Feat Possible?

- Trigonometry
- Which parabolas clear the house
- What velocity is needed on the ball, and human
- Only professional players able?
- Physical aspects
- Can it be done?

Conclusion

- Go over introduction
- Analyse findings from each chapter
- Were the aims and goals successful?
- What I learnt from this project

Appendix 2 – Maple code for Percy Grainger/Lineout

Underlining indicates changeable variable

```
> X1:=5;X2:=10;X3:=15;H:=4;L:=1;v0:=20;g:=10; #1
> sx:=X1;
> ineq:=X1*tan(t*Pi/180)-g*X1^2*(1+(tan(t*Pi/180))^2)/(2*v0^2)>H; #3
> s:=solve(ineq,t);
> ranget1:=evalf(s)[2];
> mint1:=op(op(ranget1)[1]);
> maxt1:=op(op(ranget1)[2]);
> maxx1:=v0^2*sin(2*mint1*evalf(Pi)/180)/g;
> maxy1:=v0^2*sin(maxt1*evalf(Pi)/180)^2/(2*g);
> with(plottools):with(plots):
> ph1:=display(line([X3,0], [X3,H], color=black,thickness=5)); #2
> ph2:=display(line([X1,0], [X1,H], color=black,thickness=5)); #2
> ph3:=display(line([X2,0], [X2,H+L], color=black,thickness=5)); #2
> p2:=plot({[v0*cos(maxt1*Pi/180)*t,v0*sin(maxt1*Pi/180)*t-
g*t^2/2,t=0..4],[v0*cos(mint1*Pi/180)*t,v0*sin(mint1*Pi/180)*t-
10*t^2/2,t=0..3]},0..maxx1,0..maxy1,scaling=constrained);
> display(ph1,ph2,ph3,p2);
# the code above firstly states the variables used in the equation found in the project. It
solves the inequality by giving a list of appropriate angles. It then picks the angles most
appropriate, should be between 0 and 90 degrees. It takes and gives the minimum angle as
mint1 and the maximum angle maxt1. It also gives the minimum and maximum distances
from these two angles with the variables above. It plots the graph, with the obstacles being
displayed on it. And then displays the two extreme angles going over the top of the first
obstacle.
> sx:=X2;
> ineq2:=sx*tan(t*Pi/180)-g*sx^2*(1+(tan(t*Pi/180))^2)/(2*v0^2)>H+L; #3
> s2:=solve(ineq2,t);
> ranget2:=evalf(s2)[2];
> mint2:=op(op(ranget2)[1]);
> maxt2:=op(op(ranget2)[2]);
> maxx2:=v0^2*sin(2*mint2*evalf(Pi)/180)/g;
> maxy2:=v0^2*sin(maxt2*evalf(Pi)/180)^2/(2*g);
> p3:=plot({[v0*cos(maxt2*Pi/180)*t,v0*sin(maxt2*Pi/180)*t-
g*t^2/2,t=0..4],[v0*cos(mint2*Pi/180)*t,v0*sin(mint2*Pi/180)*t-
10*t^2/2,t=0..3]},0..maxx2,0..maxy2,scaling=constrained);
> display(ph1,ph2,ph3,p3);
# the code above firstly state the variables used in the equation found in the project. It
solves the inequality by giving a list of appropriate angles. It then picks the angles most
appropriate, should be between 0 and 90 degrees. It takes and gives the minimum angle as
mint2 and the maximum angle maxt2. It also gives the minimum and maximum distances
from these two angles with the variables above. It then displays the two extreme angles
going over the top of the second obstacle.
```

```
> sx:=X3;
> ineq3:=sx*tan(t*Pi/180)-g*sx^2*(1+(tan(t*Pi/180))^2)/(2*v0^2)>=H; #3
> s3:=solve(ineq3,t);
> ranget3:=evalf(s3)[2];
> mint3:=op(ranget3)[1];
> maxt3:=op(ranget3)[2];
> maxx3:=v0^2*sin(2*mint3*evalf(Pi)/180)/g;
> maxy3:=v0^2*sin(maxt3*evalf(Pi)/180)^2/(2*g);
> p4:=plot({[v0*cos(maxt3*Pi/180)*t,v0*sin(maxt3*Pi/180)*t-g*t^2/2,t=0..4],[v0*cos(mint3*Pi/180)*t,v0*sin(mint3*Pi/180)*t-10*t^2/2,t=0..3]},0..maxx3,0..maxy3,scaling=constrained);
> display(ph1,ph2,ph3,p4);
# the code above firstly state the variables used in the equation found in the project. It solves the inequality by giving a list of appropriate angles. It then picks the angles most appropriate, should be between 0 and 90 degrees. It takes and gives the minimum angle as mint3 and the maximum angle maxt3. It also gives the minimum and maximum distances from these two angles with the variables above. It then displays the two extreme angles going over the top of the third obstacle.
> intersect_rangemin3 := proc(r1::range, r2::range, r3::range)
>   return(max(op(1, r1), op(1, r2), op(1,r3)));
> end:
>mint:=intersect_rangemin3(mint1..maxt1,mint2..maxt2,mint3..maxt3);
> intersect_rangemax3 := proc(r1::range, r2::range, r3::range)
>   return(min(op(2, r1), op(2, r2), op(2,r3)));
> end:
>maxt:=intersect_rangemax3(mint1..maxt1,mint2..maxt2,mint3..maxt3);
> maxx:=v0^2*sin(2*mint*evalf(Pi)/180)/g;
> maxy:=v0^2*sin(maxt*evalf(Pi)/180)^2/(2*g);
> pl3:=plot({[v0*cos(maxt*Pi/180)*t,v0*sin(maxt*Pi/180)*t-g*t^2/2,t=0..4],[v0*cos(mint*Pi/180)*t,v0*sin(mint*Pi/180)*t-10*t^2/2,t=0..3]},0..maxx,0..maxy,scaling=constrained);
> display(ph1,ph2,ph3,pl3);
#this is a procedure that starts by getting ready to identify ranges. It then picks the ranges from each of the three inequalities above, showing the minimum and maximum range. It then picks the smallest angle required to clear all three obstacles and the largest angle required to clear all three obstacles.
```

```
#1 changing initial conditions will let users set up conditions they want
#2 changing the H with H+L will change the location of the tall player in this model
#3 changing the inequality with the respective H or H+L will enable you to make sure the thrower is clearing the correct height
```

Appendix 3 – Maple Code for Lineout with Wind

Underlining indicates changeable variable

```
> X1:=5;X2:=10;X3:=15;H:=4;L:=1;v0:=20;g:=10;W:=-5; #1
> ineq:=X1*(v0*sin(t*Pi/180)/(v0*cos(t*Pi/180)+W))-g*X1^2/(v0*cos(t*Pi/180)+W)^2/2>H; #3
> s:=solve(ineq,t);
> evalf(s);
> ranget1:=evalf(s)[4];
> mint1:=op(op(ranget1)[1]);
> maxt1:=op(op(ranget1)[2]);
> maxx1:=v0^2*sin(2*mint1*evalf(Pi)/180)/g;
> maxy1:=v0^2*sin(maxt1*evalf(Pi)/180)^2/(2*g);
> with(plottools):with(plots):
> ph1:=display(line([X3,0], [X3,H+L], color=black,thickness=5)); #2
> ph2:=display(line([X1,0], [X1,H], color=black,thickness=5)); #2
> ph3:=display(line([X2,0], [X2,H], color=black,thickness=5)); #2
 p2:=plot({[W*t+v0*cos(maxt1*Pi/180)*t,v0*sin(maxt1*Pi/180)*t-g*t^2/2,t=0..4],[W*t+v0*cos(mint1*Pi/180)*t,v0*sin(mint1*Pi/180)*t-g*t^2/2,t=0..4]},0..maxx1,0..maxy1,scaling=constrained);
> display(ph1,ph2,ph3,p2);
# the code above firstly states the variables used in the equation found in the project. It
solves the inequality by giving a list of appropriate angles. It then picks the angles most
appropriate, should be between 0 and 90 degrees. It takes and gives the minimum angle as
mint1 and the maximum angle maxt1. It also gives the minimum and maximum distances
from these two angles with the variables above. It plots the graph, with the obstacles being
displayed on it. And then displays the two extreme angles going over the top of the first
obstacle.
> ineq2:=X2*(v0*sin(t*Pi/180)/(v0*cos(t*Pi/180)+W))-g*X2^2/(v0*cos(t*Pi/180)+W)^2/2>H; #3
> s2:=solve(ineq2,t);
> ranget2:=evalf(s2)[4];
> mint2:=op(op(ranget2)[1]);
> maxt2:=op(op(ranget2)[2]);
> maxx2:=v0^2*sin(2*mint2*evalf(Pi)/180)/g;
> maxy2:=v0^2*sin(maxt2*evalf(Pi)/180)^2/(2*g);
 p3:=plot({[W*t+v0*cos(maxt2*Pi/180)*t,v0*sin(maxt2*Pi/180)*t-g*t^2/2,t=0..4],[W*t+v0*cos(mint2*Pi/180)*t,v0*sin(mint2*Pi/180)*t-g*t^2/2,t=0..4]},0..maxx2,0..maxy2,scaling=constrained);
> display(ph1,ph2,ph3,p3);
# the code above firstly state the variables used in the equation found in the project. It
solves the inequality by giving a list of appropriate angles. It then picks the angles most
appropriate, should be between 0 and 90 degrees. It takes and gives the minimum angle as
mint2 and the maximum angle maxt2. It also gives the minimum and maximum distances
from these two angles with the variables above. It then displays the two extreme angles
going over the top of the second obstacle.
```

```
> ineq3:=X3*(v0*sin(t*Pi/180)/(v0*cos(t*Pi/180)+W))-
g*X3^2/(v0*cos(t*Pi/180)+W)^2/2>=H+L; #3
> s3:=solve(ineq3,t);
> ranget3:=evalf(s3)[4];
> mint3:=op(op(ranget3)[1]);
> maxt3:=op(op(ranget3)[2]);
> maxx3:=v0^2*sin(2*mint3*evalf(Pi)/180)/g;
> maxy3:=v0^2*sin(maxt3*evalf(Pi)/180)^2/(2*g);
 p4:=plot({[W*t+v0*cos(maxt3*Pi/180)*t,v0*sin(maxt3*Pi/180)*t-
g*t^2/2,t=0..4],[W*t+v0*cos(mint3*Pi/180)*t,v0*sin(mint3*Pi/180)*t-
g*t^2/2,t=0..4]},0..maxx3,0..maxy3,scaling=constrained);
# the code above firstly state the variables used in the equation found in the project. It
solves the inequality by giving a list of appropriate angles. It then picks the angles most
appropriate, should be between 0 and 90 degrees. It takes and gives the minimum angle as
mint3 and the maximum angle maxt3. It also gives the minimum and maximum distances
from these two angles with the variables above. It then displays the two extreme angles
going over the top of the third obstacle.
> display(ph1,ph2,ph3,p4);
> intersect_rangemin3 := proc(r1::range, r2::range, r3::range)
>   return(max(op(1, r1), op(1, r2), op(1,r3)));
> end:
> mint:=intersect_rangemin3(mint1..maxt1,mint2..maxt2,mint3..maxt3);
> intersect_rangemax3 := proc(r1::range, r2::range, r3::range)
>   return(min(op(2, r1), op(2, r2), op(2,r3)));
> end:
> maxt:=intersect_rangemax3(mint1..maxt1,mint2..maxt2,mint3..maxt3);
> maxx:=v0^2*sin(2*mint*evalf(Pi)/180)/g;
> maxy:=v0^2*sin(maxt*evalf(Pi)/180)^2/(2*g);
> pl3:=plot({[W*t+v0*cos(maxt*Pi/180)*t,v0*sin(maxt*Pi/180)*t-
g*t^2/2,t=0..4],[W*t+v0*cos(mint*Pi/180)*t,v0*sin(mint*Pi/180)*t-
g*t^2/2,t=0..4]},0..maxx,0..maxy,scaling=constrained);
> display(ph1,ph2,ph3,pl3);
#this is a procedure that starts by getting ready to identify ranges. It then picks the ranges
from each of the three inequalities above, showing the minimum and maximum range. It
then picks the smallest angle required to clear all three obstacles and the largest angle
required to clear all three obstacles.
```

#1 changing initial conditions will let users set up conditions they want
#2 changing the H with H+L will change the location of the tall player in this model
#3 changing the inequality with the respective H or H+L will enable you to make sure the thrower is clearing the correct height