# Analysis

## Background to and identification of problem

Merchant Taylors’ School is an independent school in Northwood, Greater London. It has approximately 850 pupils, from year 7 to year 13.

Sixth Form students complete A levels in their time at MTS, and are able to select subjects they wish to study as long as the head of department agrees it is sensible. In a given Sixth Form of 250 pupils, roughly half will study Mathematics in some form.

Pupils have the option to study Mathematics or Further Mathematics. Both courses are studied using the Edexcel specification. Under this specification, the Mathematics course consists of 6 modules, 4 of which are compulsory. The other 2 are selected from the areas of Mechanics, Statistics and Decision. The Further Mathematics course consists of 12 modules, 6 of these are compulsory, and again the other 6 are selected from the areas of Mechanics, Statistics and Decision.

The freedom to select modules means that a pupil studying the Edexcel specification could take a Mathematics course and not do any Mechanics Modules. However, at MTS, it is compulsory for all Mathematics students to study mechanics module M1, while it is compulsory for all Further Mathematics student to study mechanics modules M1, M2 and M3. Therefore all Mathematics students will study M1 at some point.

Considering that Mechanics is much easier modelled and understood using a computer, where the many different variables can be quickly calculated and displayed, it is surprising to find out that the Mathematics department uses no modelling software in order to aid the students with the M1 syllabus. Having spoken to many pupils, they complain that it is very hard to understand the different types of mechanical situations as there is so much to take into account, and that they find it even harder to visualize the situations they are meant to ‘fully understand’. They told me they end up learning many equations, but without being able to visualise what is going on, the meaning of the equations is often lost. This means that students make many ‘silly errors’ as they don’t really understand the concepts.

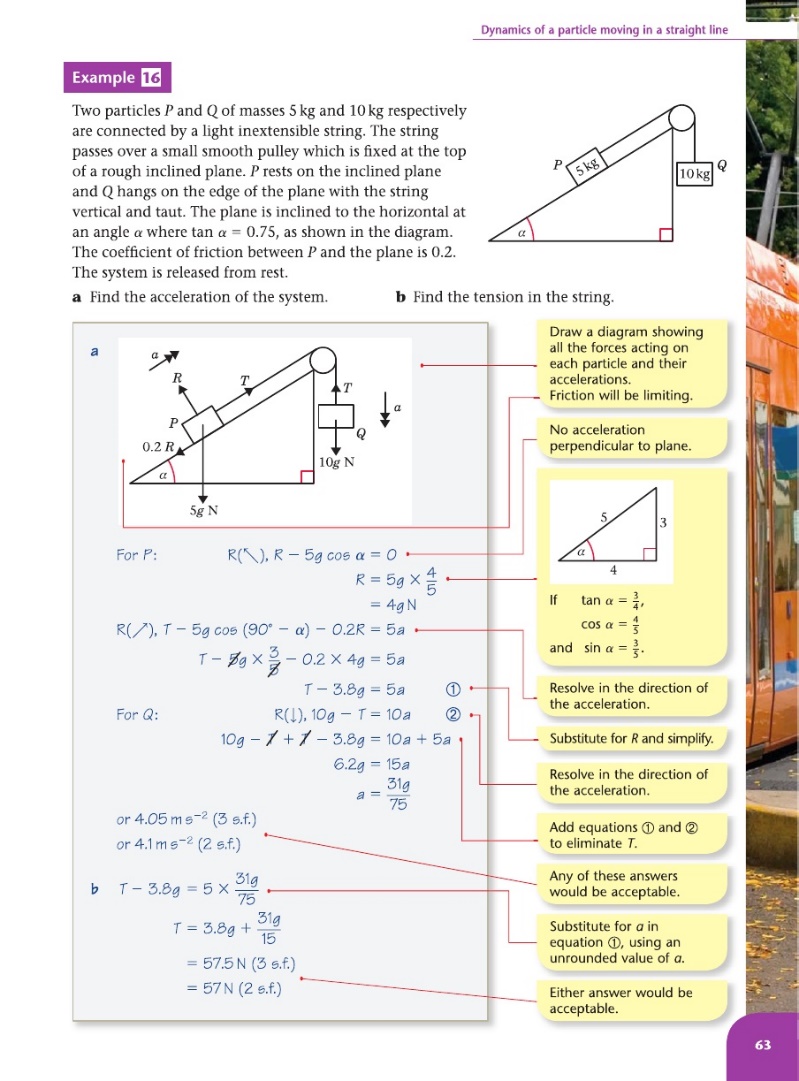
Therefore, if students were able to interact with a computer program displaying a model of a mechanical situation, it may be easier to visualize similar situations in the future, and so would be easier to solve problems involving a similar model in the future. One of the hardest parts of the M1 syllabus is ‘Pulley systems on an incline’, and so I chose to focus on this part of the course.

## Description of the current system

Currently, Pulley systems on an incline are taught using textbooks and a whiteboard. I was lucky enough to sit in on one of Mr.Slator’s (Head of Mathematics and a M1 teacher at MTS) lessons where this was being taught, and here is what I observed:

### Observations

First, the pupils were directed to page 63 of the Edexcel M1 textbook:



Terminology:

Coefficient of Friction – a number which represents how rough a surface is. The higher the number, the rougher it is.

Mr Slator then went through ‘Example 16’ in the book with the students. He explained the basics of pulley systems on an incline. Three main points which may help understand this analysis are:

1. The two connected objects are modelled as balls/particles which encounter no air resistance.
2. The incline is modelled as a 2D Plank.
3. Edexcel suggest the object on the incline is referred to as Ball 1 while the other object is referred to as Ball 2.

Following this, he made his own example on the board, which was discussed with the pupils. His example differed in that it required calculation of what would happen to the system after one of the two connected objects hit the ground. After explaining how to do this, the pupils began a worksheet to be started in class and finished for homework. The next lesson the students would move onto a new topic. I felt this was insufficient for students to truly understand the topic.

### Confirmation of Observations

Following my observation, I asked Mr Slator whether he ever used resources other than the textbook and the worksheet to teach this topic. He said that when he first taught the topic he also used a physical experiment to explain what was happening, but this was no use due to imperfections in how the system actually operated (Air resistance had an effect, for example). He accepted that the current teaching methods were probably insufficient for full understanding, but explained that he has never found a high quality resource to allow students to interact with and understand a pulley system.

Following this discussion I concluded that my initial observation was correct. This being that the current teaching methods do not allow students to truly understand pulley systems on an incline.

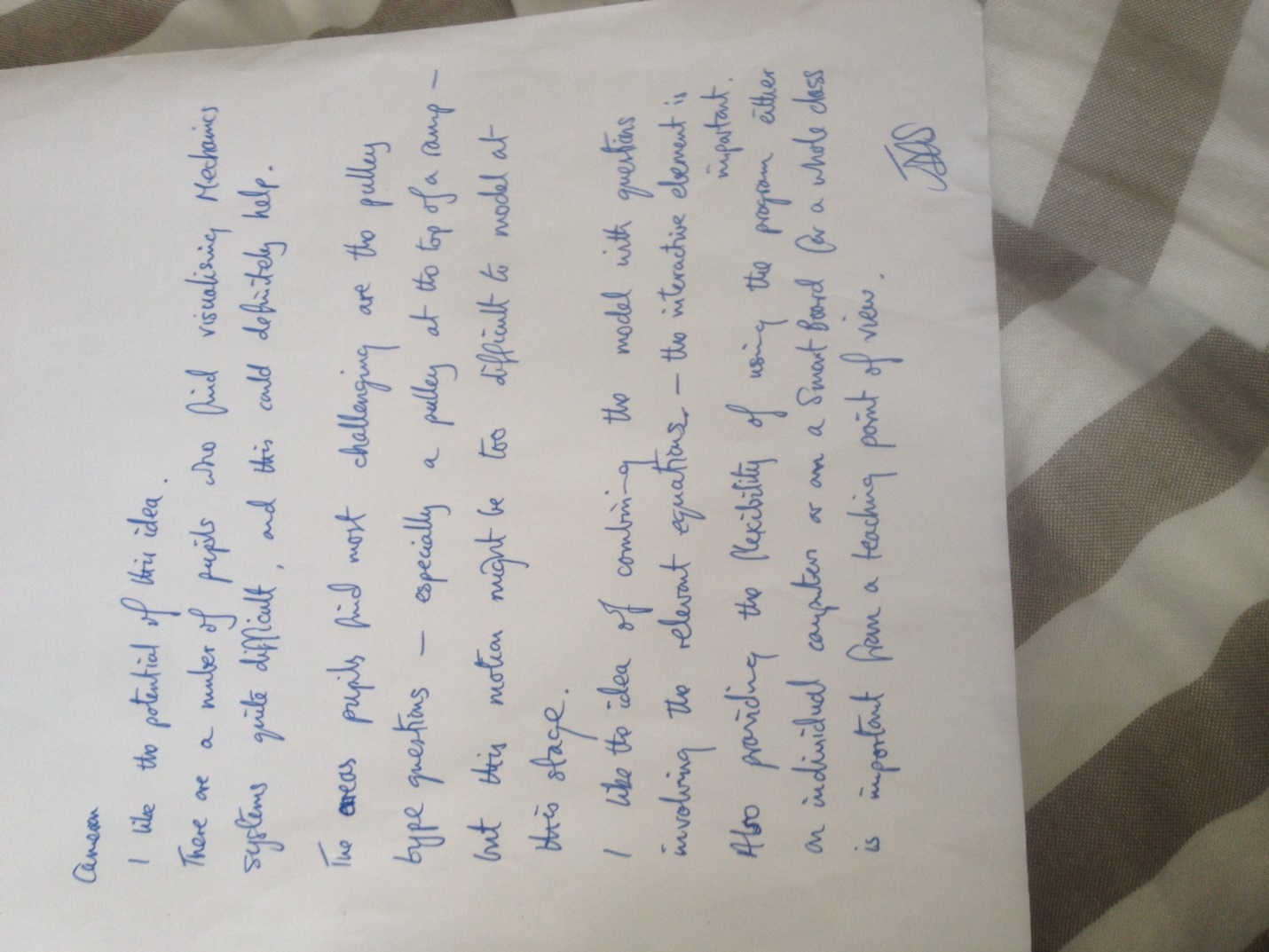
## Identification of the prospective users

Prospective users for a computerised interactive model of a pulley system on an incline would be the teachers of the M1 Mathematics module. Every year these teachers change (the staff rotate, teaching different modules). Therefore my prospective users include the Mathematics Staff at MTS and any pupils who take M1 at MTS in the future. A list of all the teachers in the Mathematics department is as follows:

* Mr Slator
* Miss Peers
* Mrs Gedalla
* Mr Rowlands
* Mr Illing
* Dr Andrews
* Mr Miller
* Mr Fothergill

## Identification of user needs and acceptable limitations

To get an initial idea of my user needs I sent Mr Slator an email explaining my idea for a computerised interactive model to help M1 students understand pulley systems on an incline. I had also suggested that the model should be accompanied by questions so the pupils can test their understanding while using the program. He gave me this written reply:



I then interviewed Mr Slator to get more information on the needs and limitations:

### Interview with Mr Slator: Maths office, 2:00pm, Wednesday 17th September 2014:

*Please could you describe the current teaching method for pulley systems on an incline?*

I first show the students an example of a problem in the textbook, which has the solution written out beneath it. After discussing this briefly, I create another example and write it on the board. This example is much trickier than the example in the book, and is what I would consider a ‘worst case scenario’ should this topic come up in an exam. After going through this problem together, I point out what variables can be ignored so we can solve these types of problems, such as air resistance. I then set the class a MEI worksheet to start in class and finish for homework.

*How easy do students find learning pulley systems on an incline?*

Students tend to find the equations relatively simple, and with a bit of luck will get most questions right. However, there is definitely a lack of general understanding and visualisation, which means that in more complex questions students tend to struggle.

*Would a program that allowed students to interact with a model of a pulley system on an incline and see an animation of what happens be helpful?*

Yes, very much so. I very much like the idea of an animation, this will keep the students engaged. Although, some systems will not move, so if this is the case please could you make the program explain why? I would also like to use the program in front of the class from time to time, so it mustn’t be limited to just the students PC’s. It must have smart board compatibility and if possible be full screen.

*Would it be helpful to have two parts to the program, one that questions students on a pre-defined model and one that allows the user to create their own model with their own variable values?*

That’s sounds very useful indeed. I would use the latter part a lot. Make sure that the part that questions students only animates once they have got the questions correct! This will encourage them to do them properly.

*For the part the questions students on the model, would you like these to be randomized?*

The questions themselves needn’t be randomized, considering that exam questions are always very similar in style. It would however be useful if the values used, such as the masses of each ball on the system, were randomized.

*Would you like to write the questions yourself? And would you like the ability to add/delete questions from a database which would in turn change the questions used in the program.*

I see no benefit in the database. In fact I wouldn’t like it at all. I would like the program to be self-sustaining, in the sense it needs no other files to be fully functional. The questions never change and I’m happy for the questions to be somehow hardcoded into the program. For the questions I trust you to do it as you have access to the textbook and past papers. I would like there to be roughly 3 questions on display per model, so it’s not too overwhelming but understanding can be properly tested.

*Ah, I understand. Are there any things that I needn’t include in the model?*

Insignificant and difficult to calculate variable such as air resistance are not required in M1. These needn’t be included.

*Would you like the ability to save, load and print different models?*

I feel that this is unnecessary, I already have the textbook and worksheets to provide me with questions.

*Do you have any other suggestions?*

I have one. Please make sure that at no point during the question any model created is beyond the M1 specification. These sorts of models can sometimes arise even when you ignore variables like air resistance.

*I’m sure that’s fine, what type of situation did you have in mind?*

If the ball falling vertically is given too large a mass, the other ball travelling up the incline will fly off the edge of the system at the top.

*No problem, I’ll make sure it handles those types of inputs. Is that everything?*

I believe so. Thank you.

*~~-~~*

From these two sets of information I have identified the following user needs:

The program must be split into two parts. Part 1 will allow a user to create a model of a pulley system using their own values, while Part 2 will present a M1 ‘pulley systems on an incline’ model and question the user on it. The model in the latter part must use random variable values to make each set of questions, while similar in style, unique. There should be roughly 3 questions per model. The program must function without the need for any other files. Both models must animate what would happen to the pulley system if it was left in its current configuration. However, the model in Part 2 must only animate once students have got all the questions for that model correct. If the system would not move and therefore result in no animation, the program must explain why. Moreover, the program must scale with screen size and be full screen so that it is usable on a teaching smart board as well as on students PC’s. Finally, the program must handle/reject any user inputs that would make the model more complex than is necessary for the M1 specification.

Acceptable limitations for the system include not needing the program to save, load and print different models. Furthermore, the models can ignore variables that aren’t dealt with in the M1 specification, such as air resistance. After the interview I came up with the idea of also creating a tablet form of the program, which may benefit some students. After discussing this idea with Mr Slator, it was decided this was not necessary. Mr Slator doesn’t think that many students would benefit from a tablet form, and would prefer me to concentrate on the PC/Smart board functionality.

## Data source(s) and destination(s)

The Data sources will be the end users (any teachers or students using the program), a random number generator and an embedded resource file. The random number generator will be used to obtain random values to use in the questions, while the resource file will allow embed a text file and image file to be embedded into the program’s executable file.

The Data destination will simply be the end users. The end users will either see a model animate, or will receive notifications that their answers to questions were correct or incorrect.

## Data volumes

The resource file will allow a text file and image file to be embedded into the program’s executable file. The image file will simply be a PNG image of a black ball. I expect this image file to be between 5-10 Kilobytes in size. The text file will store the different questions that the program will use during execution to question the user on a model. From early research, there will be approximately 10 questions that are included. These questions will be approximately 40 characters in length, and assuming 1 byte per character, this will total to 400 bytes.

## Analysis Data Dictionary (from perspective of end user)

From the perspective of the end user, these data fields will be required for processing of the models and what will happen after they are left in their configurations:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Unit | Field Type | Field Size | Example Data | Individual Validation |
| Mass of Ball 1 | Kilograms | Real | 3 digits | 25.3 | Must be positive and required digit length |
| Mass of Ball 2 | Kilograms | Real | 3 digits | 25.3 | Must be positive and required digit length |
| Coefficient of Friction between Ball 1 and the Plank/Incline | - | Real | 4 digits | 0.383 | Must be positive and required digit length |
| Angle of Plank/Incline | Degrees | Integer | 2 digits | 47 | Must be between 0 and 90 |
| Length of Plank/Incline | Metres | Real | 3 digits | 8.1 | Must be positive and required digit length |
| Length of String connecting Ball 1 and Ball 2 | Metres | Real | 3 digits | 4.2 | Must be positive and required digit length |
| How far Ball 1 starts up the Plank/Incline | Metres | Real | 3 digits | 5.9 | Must be positive and required digit length |

Note: All field sizes and field types are based on the M1 specification.

This data will either be input by the end user or randomly generated by the program based on what part of the program is being used.

There are two other variables that will need to be displayed at certain points in the program to aid the user during calculations. They could also be subject to questions.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Unit | Field Type | Field Size | Example Data | Individual Validation |
| Acceleration | Metres per second squared | Real | 7 digits | 2.456829 | None |
| Tension | Newtons | Real | 4 digits | 45.65 | None |

Note: These variables are calculated using the variables in the other table, and hence no validation is needed. Moreover, the field sizes for these two variables were advised to me by Mr.Slator. This degree of precision is needed so that the user can work backwards from these variables to accurately calculate the variables in the other table.

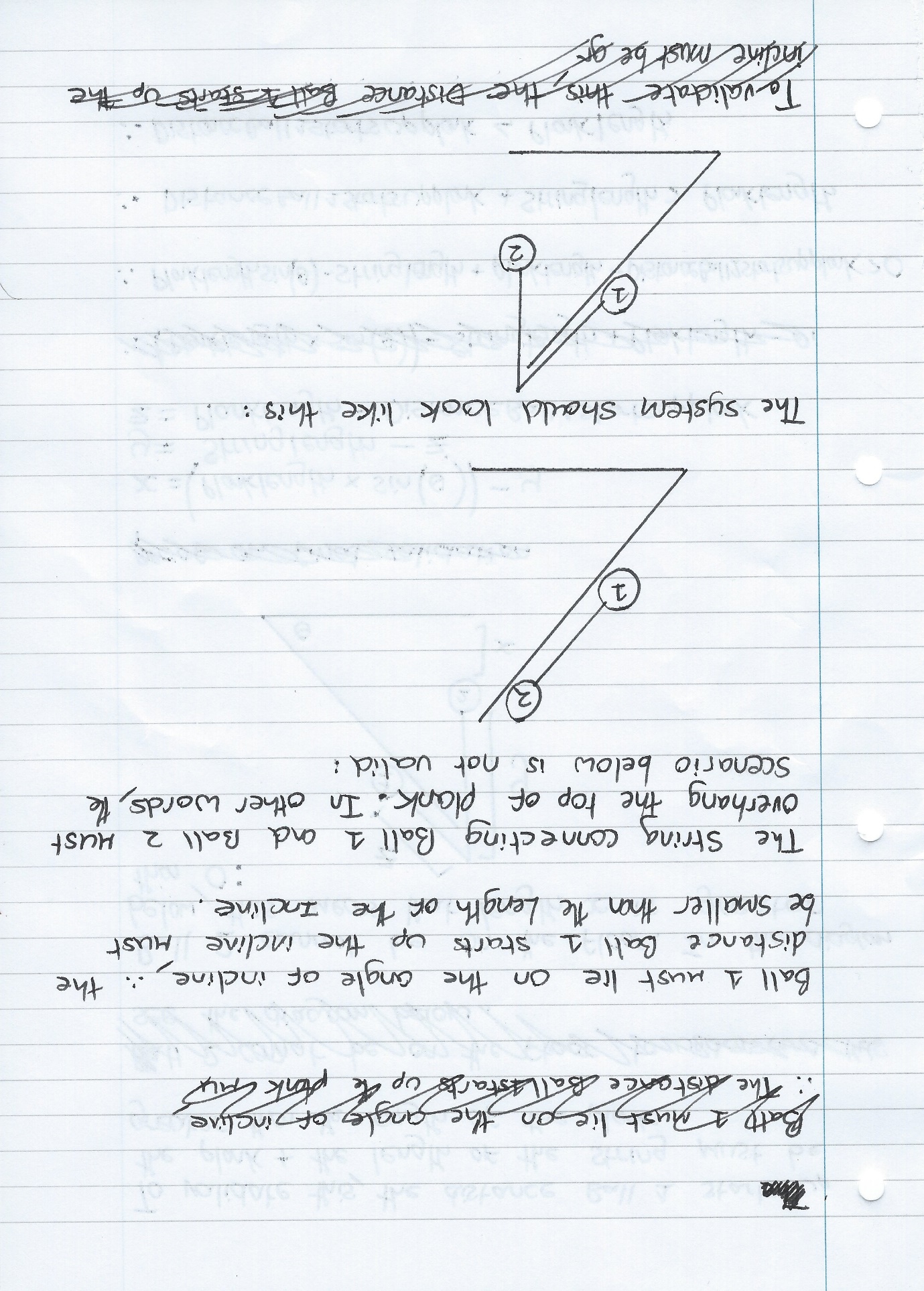
Other validation checks are also necessary to make sure that the model lies within the bounds of the M1 specification. These validation checks are found below:

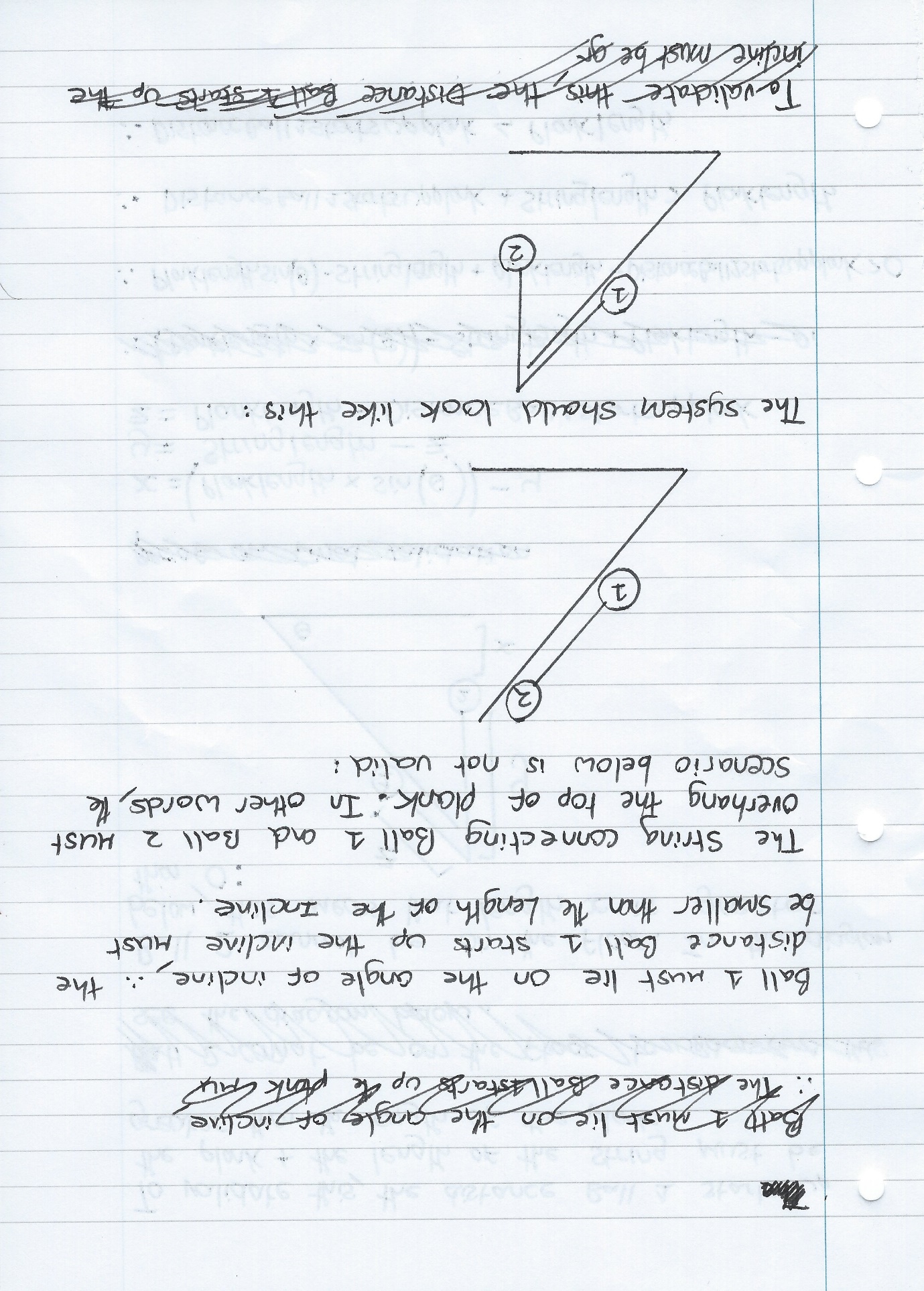
*Validation Rule 1:*

Ball 1 must lie on the incline. Therefore:

**Distance Ball 1 starts up the Plank < Length of the Plank**

*Validation Rule 2:*

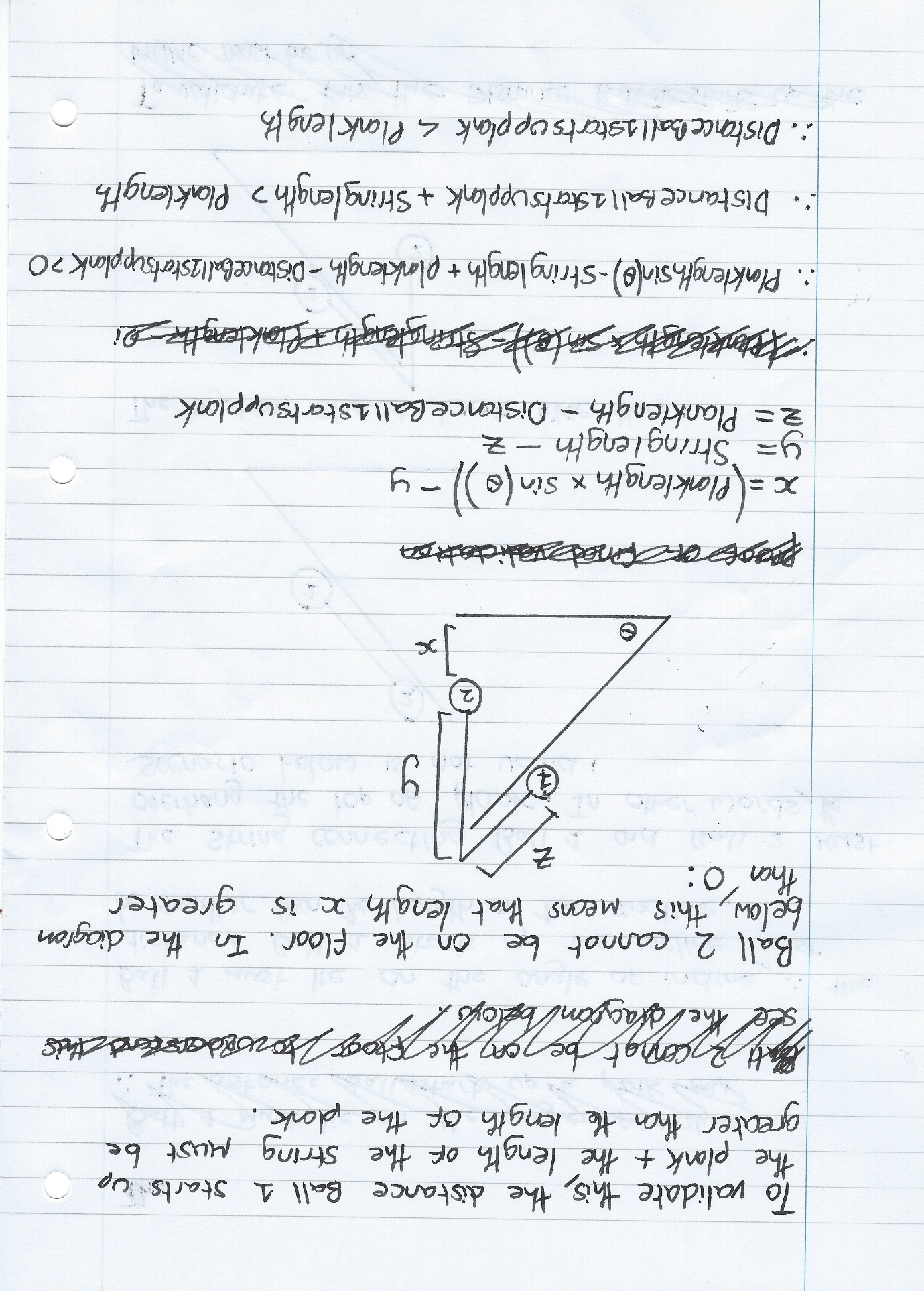
The String connecting Ball 1 and Ball 2 must be long enough to hang over the top of the plank. In other words, the scenario below is not valid:

The system should look like this:

Therefore:

**Distance Ball 1 starts up the Plank + Length of the String > Length of the Plank**

*Validation Rule 3:*

Ball 2 cannot be on the floor. In the diagram below, this means that the length X must be greater than 0:

X = Length of the Plank \* Sin() – Y

Y = Length of the String – Z

Z = Length of the Plank – Distance Ball 1 starts up the Plank

Therefore:

**(Length of the Plank \* Sin()) – Length of the String + Length of the Plank – Distance Ball 1 starts up the Plank > 0**

*Validation Rule 4:*

If moving up the slope, Ball 1 will initially move until Ball 2 hits the ground (Initial Movement), before moving a small distance (Extra Displacement) due to its previous momentum. It will then fall to its resting position.

Initial Movement can be calculated because it is the height Ball 2 started off the ground. Extra Displacement can be calculated by using the constant acceleration formula:

S = (V2 – U2)/2a

To remain an M1 problem, Ball 1 mustn’t fall off the top of the slope during this motion.

Therefore:

**Distance Ball 1 starts up Plank + Initial Movement + Extra Displacement < Length of the Plank**

This can also be written as follows if you follow through the calculations:

**(Length of the Plank \* Sin()) – Length of the String + Extra Displacement < 0**

*Validation Rule 5:*

If moving down the slope, Ball 2 will initially move until Ball 1 hits the ground (Initial Movement), before moving a small distance (Extra Displacement) due to its previous momentum. It will then fall to its resting position.

Initial Movement = Distance Ball 1 starts up Plank. Extra Displacement can again be calculated using the constant acceleration formula:

S = (V2 – U2)/2a

To remain an M1 problem, Ball 2 mustn’t reach a height higher than the top of the slope during this motion.

Therefore:

**(Length of the Plank \* Sin()) – Length of the String + Length of the Plank – Distance Ball 1 starts up the Plank+ Initial Movement + Extra Displacement < (Length of the Plank \* Sin())**

This can also be written as follows if you follow through the calculations:

**Length of the Plank – Length of the String + Extra Displacement < 0**

## DfDs (existing and proposed system) to level 1

### Existing Solution level 0

### Existing Solution level 1

### Proposed System Level 0

## 

### Proposed System Level 1

## Objectives for the proposed system

The objectives of the system/program to be implemented are as below. These were agreed with Mr.Slator following a final meeting with him:

1. To make the program consist of two parts, one which allows a user to create their own model of a pulley system on an incline and then see it animate, and another which questions the user on a randomly created model.
2. To make both parts of the program have the functionality to animate the model to show what would happen if the model was left in its configuration.
3. To make the program correctly calculate the direction of motion (if any) of a model.
4. To make the program correctly calculate the initial accleration (if any) of a model.
5. To make the program correctly calculate any extra displacement that a Ball undertakes after passing the equilibrium position due to it’s own momentum.
6. To make the program such that it is usable as a stand-alone executable without the need for any other files.
7. To have 3 questions per model in the part of the program that questions the user on a randomly created model.
8. To make the part where the user is questioned on the model only animate if the user has got all the questions correct.
9. To have the program explain why a model remained stationary if this has occurred.
10. To make sure that any animated model is within the bounds of the M1 specification.
11. To display error messages within 2 seconds.
12. To make items displayed in the program scale with screen size.
13. To make the program fullscreen.
14. To design the interface such that a user can understand and operate it with the aid of a user guide within 5 minutes.
15. To process and display randomly created models within 2 seconds of the action that triggered it.
16. To begin animations within 2 seconds of the action that triggered it.

## Realistic appraisal of the feasibility of potential solutions

Possible solutions include:

### Paper handouts of multiple examples of pulley system on an incline, with pictures

I feel like this would be an improvement on the current system, as students would see far more examples than they currently do under the current system (currently only 2). The pictures may also help the students to better visualise the questions in the future. However, the absence of questions would in my opinion make it harder for the students to remember how to solve the problems as there would be little interaction to stimulate learning. The current system at least has questions so does better in this regard. Moreover, distributing this solution to students would be expensive in material resources, such as paper.

### Already existing interactive software teaching pulley systems on an incline

One of the better solutions I found was: *http://demonstrations.wolfram.com/TwoMassesOnInclinedPlaneWithPulley.*  
This software was a fairly good solution. It allows people to interact with a pulley system on an incline from within a browser. You can tweak a lot of variables and customize the system. However, all the variables required to fulfil the M1 specification were not present (such as distance of one ball up the slope), and the software required a plugin. This plugin was rather hard to install and came with some bloat-ware. The software also does not provide any questions or animation functionality.

### Bespoke local software

This solution would allow for all the user’s requirements to be met with a system being made specifically for them. Although this would take the longest to implement, it would allow for the problems presented to most importantly:

* Be within M1 specifications
* Provide animation functionality
* Question students on models

## Justification of chosen solution

The solution selected will provide a system that will exactly match all the end-user’s requirements as there are currently no other solutions available that would allow the software to be used both as a demonstrative tool and as a self-learning tool through both user-created and randomly generated problems. This solution would be guaranteed to work on all computers within the school, scaling to the size of whatever screen is used. I had originally decided to code the system in ProcessingJS, which I had encountered on the Khan Academy website. It is very good for animations. However, as I began making a small prototype of a potential solution I found that it wasn’t as good for event-driven programming. This was something that I deemed very important as I would like the program to be as interactive as possible. To this end, I have chosen to code the system in Delphi-Firemonkey due to its ability to create forms with ease, it is event-driven and it allows for a compiled executable to be created with an embedded resource file.

## Identification of Objects and Analysis diagrams for Object-Oriented Programmed solutions

I intend to use many of the Objects that Delphi has built into it. Such as TStringList and TResourceStream. However, there is one class that I intend to self-define. This class will be ‘TBall’, Ball 1 and Ball 2 being membesr/objects of this class. TBall will inherit from the TImage class in Delphi as follows:

TBall

TImage

I will override the Create and Destroy TImage methods so that the Balls can be displayed using my own embedded image file. Inheriting from TImage will also give me access to the Balls’ positions and the animate method. It was not an option to just make Ball 1 and Ball 2 members of TImage is because you cannot independently animate two TImages objects at the same time in Delphi-Firemonkey.