DESIGN FOR ASSEMBLY: PART ORIENTATION, PREDICTION, AND DEMONSTRATION

A Major Qualifying Project Report:

Submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

in

Mechanical Engineering

By

____

Pitchaya Kuntanarumitkul

____

Ni Pan

____

Jason Zelle

Date: 4/30/2015

Project Advisor: Cosme Furlong

Project Co-advisor: Diana Lados

Abstract

Assembly manufacturing machines require components to be fed to the work head in the correct orientation. To minimize assembly time, parts exiting the feeders have to be consistently oriented, which is not always achieved. This paper reports efforts made in creating theoretical and computational methodologies and tools to predict percentage occurrences of final orientations of parts being dropped into feeders. The theoretical developments are based on the surface areas and the location of the center of gravity of a part without accounting for its elasticity while the computational developments, based on Finite Element Analysis, account for the deformation of a part, its material properties, dropping height, velocity, and dropping angle. The end results of this project are two separate software tools that have been validated by physical testing, which can be used to predict the percentage occurrence of different landing orientations. In addition, the tools enable the ability of a designer to manipulate dimensions and material properties of a part, so that it has a desired landing orientation to potentially increase efficiency of assembly.

Acknowledgments

The completion of this project would not have been possible without the support of many individuals. The team would like to extend our gratitude to the individuals that helped this project possible. Thank you Professor Cosme Furlong for advising the project and providing useful guidance. Thank you Professor Diana Lados for co-advising the project and thank you Koohyar Pooladvand for teaching us how to use the high-speed imaging system, and for helping us take the footages of the product dropping. We would also like to express our genuine appreciation towards Loren Gjata and Brian Guerette for overseeing the project and providing resources and feedback to the project.

Executive Summary

The final orientation of a part is critical for manufacturing processes. Parts are usually fed into the feed systems in bulk and the feed systems ensure that only correctly oriented parts are fed to the work head. Often, undesired orientated parts are rejected and only parts with the desired orientation are allowed to pass through the production line. The feed rate of the feed systems is dependent upon the percentage of parts that naturally come to rest in the orientation desired by the work head. Thus, when a part is correctly oriented as it is fed into its initial phases of a manufacturing line, better production efficiency can be achieved. When designing a feed system, knowledge of the natural resting orientations of a part when dropped is critical. Currently, the natural resting orientations of a part being dropped are found by using physical testing. This method requires the manufacturing of the prototypes, which is costly and time consuming. Therefore, a virtual model that can predict the natural orientation of parts with complex geometry is desired by the sponsor.

This project focuses on creating theoretical and computational methodologies and tools to predict percentage occurrences of a part's final orientations when being dropped, thus providing the sponsor with both a mathematical and a computer model for part prediction. The results from the two models were validated using physical testing.

The theoretical portion of the project is based on a current predictive model called the Stability Method. This method takes into account the surface area and the location of the center of gravity. The method was applied to simple and complex parts and validated through physical testing. The original method does not provide accurate results when applied to parts with complex geometries, thus the method was modified. The modification of the method was conducted in two

main stages, the first being called the Modified Stability Method and the second being called the New Modified Stability Method. The Modified Stability Method takes into account geometrical features involving curved surfaces and holes. The New Modified Stability method develops on the previous methods while also taking into account surfaces that the object can land on but cannot stabilize on. Chi-square test for independence was conducted on the data obtained from physical testing and the data obtained from the application of the different stages of the Stability Method to determine if the difference in values shown is expected or due to chance. For each of the calculations, the confidence level was 0.9 and the confidence interval was 0.005. The difference in values found from the application of the New Modified Stability Method to the Simplified model of the product was considered not significant and it can be said with 90% confidence that the difference in the data is most likely due to chance. The utilization of the program involves the use of SolidWorks and a MATLAB script developed by the team. SolidWorks is used to find the values of the surface areas and their distances from the center of gravity, while the MATLAB script is used to compute the mathematical model and provide a histogram plot of the different percentage occurrences of each landings surface. The script is also able to provide alteration suggestions to the given design.

The computational portion of the project utilizes Finite Element Analysis (FEA) software, ABAQUS and investigates the parameters that may have an effect on part orientation such as its material properties, dropping height, velocity, and dropping angle. This portion involves the use of a Python script, which automates the process and eliminates any direct interaction with ABAQUS from the user, and the use of a MATLAB script to extract the data obtained from the FEA program. For the computational portion, the part is meshed using ABAQUS and convergence tests were conducted to ensure accuracy of the solution. The data obtained from the program was compared to the data obtained from physical testing to give an average of error 14.5% for parts with simple geometries and a 2.8% error for the part of interest. Thus, the tool is able to give part orientation predictions. However, the use of the tool requires a large computation time and cost.

The end results of this project are two separate tools that can be used to predict the percentage occurrence of different landing orientations. The tools can be used in the design and development of feeder systems. In addition, the tool will allow designers to manipulate dimensions and material properties of a part so that it has a desired landing orientation without having to physically manufacture the prototype, this reduces the cost of development. Having the desired landing orientation increases the maximum feed rate and therefore the efficiency of the assembly.

Table of Contents

1. Introduction

On an assembly line, with machines such as bowl feeders, correct part orientation is critical to the overall system efficiency and production rate. Feeding in parts with undesired orientation will not only increase the manufacturing time, but also increase the overall manufacturing cost for the company. This project focuses on dealing with this challenge. Methodologies and tools in predicting a part's final landing orientation after dropping were researched and developed for our sponsor to utilize.

1.1. Objective

The objective of this project is to provide a reliable tool that is able to predict the probability of each landing surface of a part. The tool will be a virtual model that can predict the natural orientation of parts, having complex geometry, when dropped on a surface. The project puts emphasis on the development of theoretical and computational methodologies and tools to predict percentage occurrences of the final orientations of parts being dropped. This knowledge is critical in the manufacturing process as parts that are fed into the work head are rejected if they do not have the desired orientation. Thus, when parts are incorrectly orientated, the efficiency is decreased.

1.2. Importance of Project

In an assembly line the orientation of a part is extremely important, without the proper orientation in most cases the part is not allowed to move forward through the assembly. Therefore controlling the orientation of the part becomes one of the main goals for any manufacturing process. For this project our sponsor is looking to improve the feed rate of parts passing through a vibrating bowl feeder. One way to do this is to alter how the parts land in the bowl feeder. If the landing probabilities are shifted to one closer to the final outcome of the bowl feeder, it will cause less parts to be rejected and in return increase the rate at which parts enter the manufacturing line. When the rate of production of a part is increased it will lead to reduced manufacturing cost and therefore greater profit margins.

1.3. Methodology Overview

The methodology portion of this paper is divided into three main parts: the development of the theoretical method, the development of the computational method, the automation of simulations and the experimental testing conducted.

The theoretical portion discusses the different existing predictive models and develops on the Stability Method. The section deliberates the limitations to the original Stability Method and experiments with modifying the original method and applying it to complex geometries, including a simplified version of the product of interest.

The computational portion discusses the manual procedures of simulating a part dropping process using finite element method and the process of automating FEA procedures. The section also elaborates on the procedures in meshing, and the parameters such as dropping velocity, height and angle that need to be considered in the computational method.

The automation of simulations portion discusses the different methods and programs involved in automating the computational methodology and how the script was developed.

The experimental testing portion discusses the procedure conducted to carry out the experiments and the work conducted using high speed imaging cameras.

2. Background

2.1. Assembly Lines

The idea of the assembly lines comes from the scientific revolution of the eighteenth century when they tried to quantify and find ways to make an industry more productive. The goal was to create an industry that functioned without human labor. The idea was not realized until nineteenth century. In 1899 Henry Ford started his own automobile manufacturing company and broke the tasks involved in the manufacturing of an automobile down to the function of each autoworker, conceiving of each worker as a part of a machine that made cars (Goss, 2010). During that time, assembly line not only saved the manufacturing time dramatically, but also increased company overall profit and workers' wages.

Therefore from a basic point of view, an assembly line is a series of stations at which people or machines add to or assemble parts for a product. It is an industrial arrangement of machines, equipment, and workers for continuous flow of work pieces in mass-production operations. The design for an assembly line is determined by analyzing the steps necessary to manufacture each product component as well as the final product. All movement of material is simplified, with no cross flow or repetitious procedure. Work assignments, number of machines, and production rates are programmed so that all operations along the line are compatible.

An assembly line (Assembly Line, 2003) can begin with a number of different lines, each devoted to a different component of a product. The lines converge upon one another, until only one line is left, producing the final product. For example in automotive companies, assembly lines usually begin with raw materials and end five miles away with a completed automobile. The structure for a relatively complex assembly line begins as one main line with stations along it that are fed by lines running perpendicular to it, with each of these side lines feeding components for the finished product.

One of the values of the assembly line is its versatility and their wide range in complexity. In addition, it can lower the cost of labor because the system is heavily reliant upon machines and requires little to no human supervision.

2.1.1. Bowl Feeders

A large number of modern day manufacturing firms use part feeders in their manufacturing line. Part feeders are machines that orient parts and feed parts to other processes, machines or conveyors (Jaksic & Maul, 2001). Feeders are commonly designed to orient a single part so that each part has a specified orientation upon leaving the feeder. Parts that do not have the correct orientation are rejected back to the feeder. These machines are responsible for almost one third of the cost and failure risk associated with the manufacturing line and the assembly system (Berkowitz & Canny, Designing Parts Feeders Using Dynamic Simulation, 1996).

Bowl Feeders are used in industrial production lines to align and feed individual components in automated assemblies. They are created to orient specific components into the desired orientation. There are two main types of bowl feeders, vibratory and non-vibratory (Jaksic & Maul, 2001). An example of a bowl feeder in an industrial production line is the system of sorting and bottling pills. In this case, part orientation is not imperative as the bowl feeder in the system is used to restrict the number of pills that can leave the feeder at a time so that they can be counted. The feed stops once the correct number of pills is in a bottle and continues once a new bottle is in its place.

The designs of bowl feeders are specific to the component in the assembly line. Thus, the designing process is very time consuming. Vibratory bowl feeders used today were developed to be used with specific parts. Vibratory bowl feeders cannot be developed based on theory and automatic designs. Rather, they are developed by modifying previous designs and by empirical debugging (Berkowitz & Canny, Designing Parts Feeders Using Dynamic Simulation, 1996). One of the issues with the feeders used for feeding and orientating parts for assembly is that objects have more than one stable configuration.

Each bowl feeder is designed to orient components in a specific orientation. Typically, bowl feeders are mounted on a base by three to four leave springs (Maul & Thomas, 1997), these leaf springs act as legs to support the bowl. A rubber foot is mounted under the leaf springs to act as dampers against external vibrations. This is shown in Fig. 1.

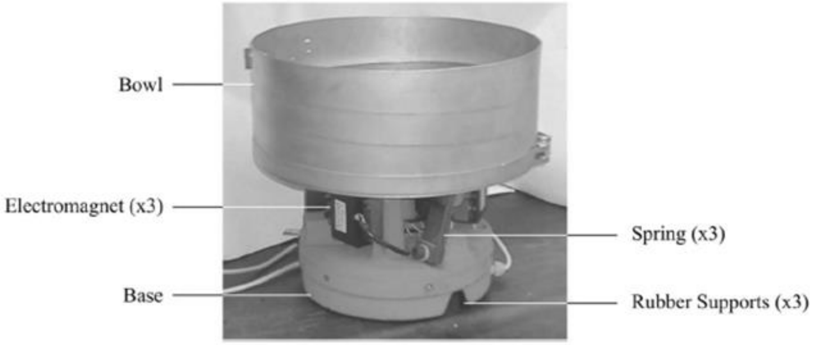


Fig. 1. Layout of Vibratory Bowl Feeder (Silversides, Dai, & Seneviratne, 2004)

Bowl feeders consists of a helical track on the inside wall of the large metal bowl. The components are delivered in bulk in an uncontrolled fashion, thus entanglements between parts are common. The components enter the bowl feeder from the top, then rest at the bottom of the bowl, and components are individually forced to move up the circular track from the vibration of the bowl. The vibration of the bowl can be produced either by electromagnetic or pneumatic drives, both of which are controlled by a drive unit. The drive exists to ensure that the objects moving keep up the bowl feeder. Tracks on bowl feeders are specifically designed for a component, thus tracks on bowl feeders may have different protrusions and floor cutouts to orient parts in stages (Berkowitz & Canny, Designing Parts Feeders Using Dynamic Simulation, 1996). The parts can be oriented in two ways, passive or active. The passive method involves rejection of parts that are not in the desired orientation. The rejected parts are then guided off the track and returned to the bottom of the bowl (Jaksic & Maul, 2001). The active method involves manipulating the part so that it has the desired orientation instead of returning it into the bowl. This method is the preferred method as it increases the feed rate and reduces possible damages to the part (Jaksic & Maul, 2001).

2.2. Effects on Final Resting Orientation

Orientation is a vital part to any assembly process. Various methods are carried out to correct any improperly oriented part. If the parts were to land in a more optimal orientation in the beginning of the assembly, it would drastically increase the speed at which a bowl feeder or other methods can carry out their task. This would lead to increased production rates, which will lead to decreased cost of production and therefore an increased profit. Since the end goal of most business is increase their profit finding a means to increase production rate would be ideal for any business. In order to change the spread of a part's landing orientations either the method of how the part is added to the assembly line or the geometrical and material properties of the part itself. .

With respect to part geometry, the two main factors that come into play are the center of gravity and the size of the landing surface (Suresh, Jagadeesh, & Varthanan, 2013). This article states that the final resting orientation of a rigid part is only dependent of the surface area and distance that surface is from the center of gravity. This is assuming that the part is dropped from a sufficient height. This means that the closer the center of gravity is to a face, the more likely a part will land on that face. In addition, the larger the face is the more likely a part will land on it. However this was proven to only hold true for rigid parts with simple geometry that are made from a single material.

The conditions, which a part is dropped, can also affect the landing orientation; however the main impact of the final resting orientation of a part is due to its geometry and material properties. For the scope of this project the following conditions were considered: drop height, drop angle, initial dropping velocity, and initial orientation.

2.3. Drop Testing

Drop testing is very common among handheld and mobile products including smart phones, radios, and remotes. Typically it is used to assess the fragility of electronic products. Nowadays these products are becoming increasingly complex and are subjected to more abuse than they have ever before. Therefore in order to ensure their functionality, the company needs to perform drop testing on the products by subjecting them to repeatable impacts similar to those they will experience with an end user before a new device hits the market. Usually drop testing is performed on small (< 1kg) products that are either used in the hand, or in mobile situations (Halt & Hass, n.d.).

A critical component in a drop test is control over repeatability (Halt & Hass, n.d.). For example, if the company is making product comparisons through drop testing, the results would not be a reliable tool for comparison unless the drop height, initial dropping orientation and dropping impact point are kept constant. This type of repeatability control is hard to achieve by human beings, thus a lot of drop testing machines are designed to fulfill this task.

Different dropping testers are available on markets nowadays. They usually have the ability to control the drop height in varying orientations and velocity. However, the product’s final condition is what most companies are interested in. It is an inefficient process since the products needed to be inspected by a person after each drop. In addition, the final impact on a product structural integrity may be hard to analyze just through its appearance.

Another method used to perform drop tests is the use of simulation software. This method provides several advantages over utilizing drop testing machines. First, it reduces costs for manufacturing of the prototype. The physical products must be available before performing drop testing in a tester. If the company is interested in finding out how different features may affect a product’s dropping performance, all the parts need to be prototyped and then tested one by one. However, with the use of simulation software these parts can be modeled using CAD tools with no additional costs. In addition, the part being tested can be subjected to real-world conditions in the simulation software. One can also find out how dropping a product will affect its structural integrity quickly and efficiently and understand its impact strength to ensure an adequate service life. The drop can be easily controlled and repeated through simulation, thus eliminating rework and saving time and development costs.

Many of the performance characteristics of a product are clearly evident when evaluated using a drop testing. However, the speed during a drop test in many cases is too fast for the human eye to discern. High speed imaging techniques can be employed in such scenario, which provides the possibility of slowing down the event, taking images at 5000 frames per second and faster. This allows the user to clearly identify all the characteristics of performance of a product in a visual manner.

2.4. Theoretical Predictive Models

In order to gain a better understanding of existing knowledge and technology in the field, research was conducted on different methods that can be used to predict part orientation. The predictive models are purely mathematical and used to predict future outcome using statistics, given a set of input data. Three existing part orientation predictive modeling methods were explored in detail. These methods are: Stability Method, Centroid Solid Angle Method and Energy Barrier Method (Suresh, Jagadeesh, & Varthanan, 2013). The details and analysis of the aforementioned methods can be found in Section 3.2.1. in the Methodology portion of this paper.

2.4.1. Limitations

Theoretical predictive model are purely mathematical and theory based, the method does come with limitations. First, the existing part orientation predictive modeling methods do not take into account the elasticity or deformation of the object and only applies to rigid parts with no consideration to material properties. Moreover, it does not take into account the initial part orientation, speed and angle of drop and does not take into account the initial angle of impact. This method can only predict the statistical outcome of the final resting face of the object. Furthermore, computation may be complicated for objects with complex parts or curved surfaces.

2.5. Computational Modeling

Researches on ways to simulate the part were conducted to eliminate the need to produce the parts for physical testing. One of the leading methods for physical world simulations is finite element analysis (FEA). FEA is a method used to solve complex physics based scenarios via computer software. The main uses of FEA in industry can range from simulations of fluid flow to impact and stress calculations. FEA is also the main method that many companies use to conduct drop tests on products before producing them. However these drop tests are mainly used to test the parts’ strength and durability and not to predict its landing orientation.

The main benefits from an FEA based analysis is its ability to take into account the conditions that the part undergoes during assembly. FEA takes into account complex geometries and material conditions along with all of the initial conditions that a theoretical based method does not account for. Some of the main FEA software on the market at the moment are, LS-DYNA, COMSOL, ANSYS, and ABAQUS. Some of the factors that came into play when choosing the software was the, availability, strength of contact modeling, and the ability to combine with a programing language to automate the results.

3. Methodology

3.1. Introduction

The methodology section of this paper discusses two methods that can be used to predict part orientations. The first method is a theoretical method based on equations and probabilities, while the second method requires the use of Finite Element Analysis (FEA) software.

3.2. Theoretical Method

The theoretical portion involves research and analysis on existing predictive models, the selection of the method and how the method was developed to meet the objective of the project.

3.2.1. Existing Theoretical Predictive Models

Research was conducted on three different existing theoretical methods including: (a) the Stability Method; (b) the Centroid Solid Angle Method; and (c) the Energy Barrier method. Unless otherwise stated, the method employs the same surface type assumption: the surfaces of a part are either hard or soft. A surface is considered soft if it rolls across the surface it was resting on when a part is dropped on it and a surface is considered hard if the impact force from an object is being dropped on it does not cause the part to roll across the surface. (Ngoi, Lim, & Ee, 1993).

a. Stability Method

The Stability Method is based on features effecting part stability. Stability is a function of "the size of the contacting surface area and the distance of the center of gravity from the base.

Stability of the orientation is directly proportional the surface area and inversely proportional to the height of center of gravity" (Suresh, Jagadeesh, & Varthanan, 2013). The theory takes into account the area of each surface and the location of the center of gravity relative to each surface, thus the material and shape distribution.

The theory can easily be understood by thinking about the relationship logically; having large surface areas in contact with the ground creates more stability for the object. Moreover, the closer the center of gravity is to the ground, the more stable an object is. The probability of each orientation of an object can be determined by using the equation below:

$$P_i = \frac{\frac{n_i\, A_i}{y_i}}{\sum \frac{n_i\, A_i}{y_i}}, \qquad (1)$$

where, P_i stands for the probability of the object landing on orientation 'i', n_i is the number of surfaces identical to and inclusive of the contacting surface 'i', A_i is the contact surface area of orientation 'i' measured in millimeter-squared (mm^2) and y_i is the distance of the contact surface from the center of gravity measured in millimeters (mm) (Udhayakumar, Mohanram, Keerthi Anand, & Srinivasan, 2013).

Fig. 2 shows the three natural resting orientations of a regular rectangular prism, resting on surface 1, 2 and 3 respectively. A regular rectangular prism has three unique landing surfaces and six surfaces in total. Each of the unique landing surfaces has a surface that is identical to itself. These surfaces are highlighted in Fig. 2. The application of the Stability method to the rectangular prism will result in three unique probabilities respective to each unique landing surface. The surface area and distance from the center of gravity values that are used to calculate the probability of landing on each unique surface is shown on the diagram. In this case, the number of surfaces identical to and inclusive of the contacting surface (n) is two for all of the unique landing surfaces because each unique surface has one other identical surface.

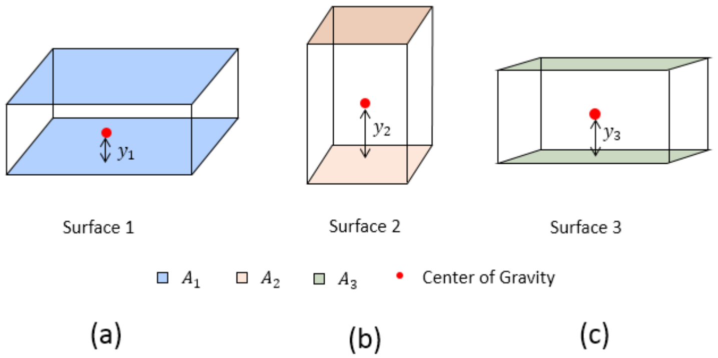


Fig. 2. Stability Method Cuboid Example showing number of surfaces, surface area and distance from center of gravity: (a) Surface 1; (b) Surface 2; and (c) Surface 3.

b. *Centroid Solid Angle Method*

The Centroid Solid Angle method is based on the hypothesis that the probability of an object landing on a particular surface and stabilizing at a resting orientation is "directly proportional to the solid angle or solid angle ratio subtended by the centroid to that surface, and inversely proportional to the height of the centroid from that resting orientation" (Suresh, Jagadeesh, & Varthanan, 2013). A "solid angle subtended by a surface is defined as the surface area of a unit surface covered by the projection onto the sphere… A solid angle is measured in steradians" (Weisstein, n.d.). This method is usually applied to objects with irregular cross sections.

The hypothesis can be written in terms of a generalized equation described as

$$P_i = \frac{\frac{W_i}{h_i}}{\sum_{j=1}^{n} \frac{W_j}{h_j}}, \tag{2}$$

where P_i is the probability of the object landing on orientation 'i', W_i is the centroid solid angle of orientation 'i' and h_i is the height of the center of gravity from the landing surface in millimeters (mm) (Udhayakumar, Mohanram, Keerthi Anand, & Srinivasan, 2013).

The equation for the centroid solid angle is

$$W_i = \frac{A}{R^2}, \tag{3}$$

where A is the surface area and R is the radius of the solid angle generation. These two values are obtained from the enveloped volume, which is discussed later in this Section.

The solid angle can be generated by connecting the vertices projected from the predicted landing surface with the center of gravity. Fig.3 shows an example of vertices projection on a cuboid.

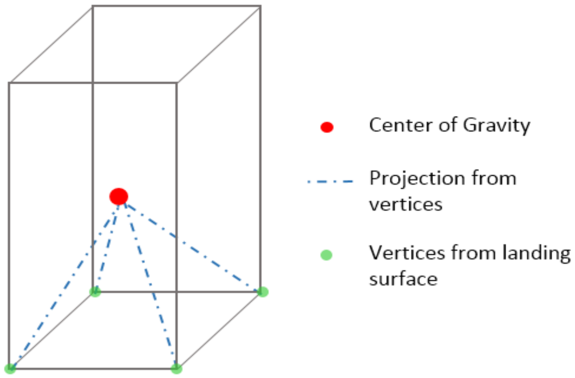

● Center of Gravity
-·-·- Projection from vertices
● Vertices from landing surface

Fig.3. Construction of Solid Angle, example of vertices projection on a cuboid

The vertices of the projected surface are connected to the center of gravity to create a prism. The next step is to construct a sphere with an arbitrary radius, R with the center of gravity as the center. This is shown in Fig.4. The intersection between the solid sphere and the prism is called the enveloped volume.

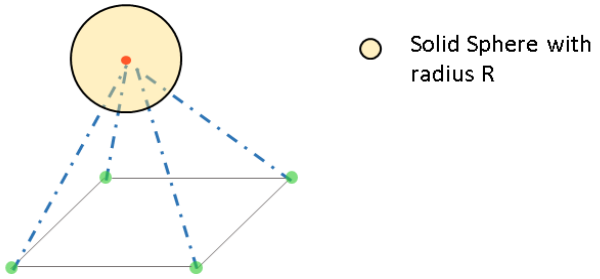


Fig.4. Construction of Solid Sphere on the projected vertices of a cuboid with the center of gravity as the point of origin

The geometry shown in Fig.5 is the enveloped volume. The radius, R and the surface area A is used to calculate the Centroid Solid Angle based on Equation 3.

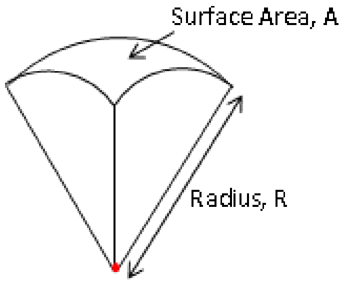


Fig.5. Enveloped Volume found from the intersection of the Solid Sphere and the projected vertices

c. Energy Barrier Method

The Energy Barrier Method is based on the assumption that the "probability for a part to come to rest in a particular natural resting orientation is a function of two factors namely, the energy tending to prevent a change of orientation and the amount of energy possessed by a part when it begins to fall into that natural resting orientation" (Suresh, Jagadeesh, & Varthanan, 2013). In theory, this method should be capable of analyzing all types of geometries. The probability of each orientation occurring can be found from the equation below:

$$a = \frac{x_a}{x_a + x_b + x_c}, \tag{4}$$

where a stands for the probability that the object will land on surface a, x_a stands for the energy barrier of aspect x_b and x_c and stands for the energy barrier of aspects b and c respectively (Udhayakumar, Mohanram, Keerthi Anand, & Srinivasan, 2013).

The numerical calculation of the energy barrier is conducted in multidimensional space. This involves the search for the minimum energy path, a continuous path that joins two minima. Thus, there is no easy way to calculate the energy barrier.

3.2.2. Selection of the Theoretical Method

The aim of this project is to provide the sponsor with a simple and user-friendly tool that is able to give the probability of an object landing on a particular surface. Thus, one of the most important aspects to consider is the simplicity of the application of the method. In addition, the accuracy of the method has to be taken into consideration.

In theory, the energy barrier method should be able to accurately predict the probability for a part to come to rest on a particular surface for any object. However, the method is computationally intensive as it requires computation of energy paths. Thus, the method is not suitable for our application.

The accuracy of the Centroid Solid Angle Method and the Stability Method can be compared based on previous studies, found in Appendix A. The study compares the results obtained from the Centroid Solid Angle Method versus the Stability Method of eight different objects. The graphs shows that the results found from the two methods very closely resembles each other, thus assuming the theories are accurate, then either of the methods can be selected for further investigation.

The Centroid Solid Angle Method and the Stability Method both involve the use of a CAD program to find the values of the variables needed to compute the part’s probabilities. The Centroid Solid Angle method involves finding the distance between the landing surface and the center of gravity. In addition, it involves the construction of the solid angle as well as the separation of the enveloped volume to find the surface area needed to compute the value of the centroid solid angle. The stability method involves finding the surface area of the landing surface as well as the distance between surface and the center of gravity. The difference between the two methods in terms of the variables needed to be obtained from the CAD program is the surface area considered, the surface area of the landing surface can easily be found by using the surface area measurement tool in most CAD programs while the Centroid Solid Angle will most likely involve the construction of the solid angle and the solid sphere by the user before the surface area measurement tool can be used.

Since the most user friendly method is desired, the Stability Method was selected for further analysis and evaluation.

3.2.3. Determining the Limitations to the Original Stability Method

In order to validate the Stability Method, the method was applied to simple and complex parts. The method was applied to a rectangular prism to prove that the method works with simple parts. The theoretical result for each object is compared to the result obtained from physical testing, which unless otherwise stated, were all conducted by the team. The procedure for the physical testing can be found in Section 3.5.1. For each object, the number of drop tests conducted exceeded the suggested sample size required detailed in Appendix D. Sample Size Calculation. In addition, the Chi-square test for independence was conducted on the two sets of results (theoretical and physical) to determine if the difference in values shown is expected or due to chance. Appendix E discusses in detail the procedure to carry out a Chi-square test for independence and the different variables considered in the calculations. Table 17, found in Appendix E shows the chi-square value and the p-value obtained by comparing the physical and theoretical results. For each of the calculations, the confidence level was 0.9 and the confidence interval was 0.05.

a. Application to Rectangular Prism

The Stability Method was used to predict probabilities of the landing orientations of a regular rectangular cuboid. The rectangular cuboid was modeled in SolidWorks and the program was used to find the surface area of the landing surfaces and the distance of the surfaces from the center of gravity.

The results obtained from the numerical values from SolidWorks were compared to the results from an article by B.K.A Ngoi, L.E.N Lim and J.T Ee, that gives the physical testing results from drop tests conducted on square and rectangular prims with varying aspect ratios. The prisms were made of aluminum squares and were tested on hard and soft surfaces. The results from the paper can be found in Appendix C. Physical Testing Results for Rectangular Prism as described in Section .

The rectangular prism was modeled with aspect ratio length to width (*y/x*) of 0.56. Fig. 6 shows the aspect ratio used to model the rectangular prism and the different surfaces.

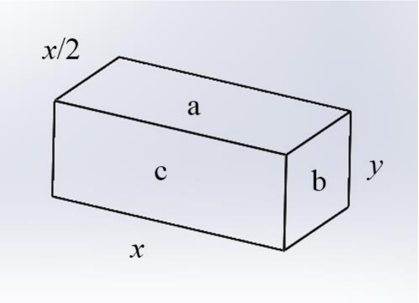


Fig. 6. Rectangular Prism modeled in SolidWorks showing the length, width and surfaces.

An arbitrary value for *y* was selected and *x* was calculated to give the length to width ratio of 0.56. The numerical values of the area of contacting surface and the distance of the surface from the center of gravity obtained from measurement tool in SolidWorks and the values of *y* and *x* used to dimension the model is shown in the Table 5 in Appendix B. Values of Variables considered. The values were applied to the Stability Method equation to find the theoretical probability of the part's landing surface. The percentages obtained from the equation were compared to the results from the theoretical value and the two sets of data were plotted in a histogram, which can be found in Fig.7.

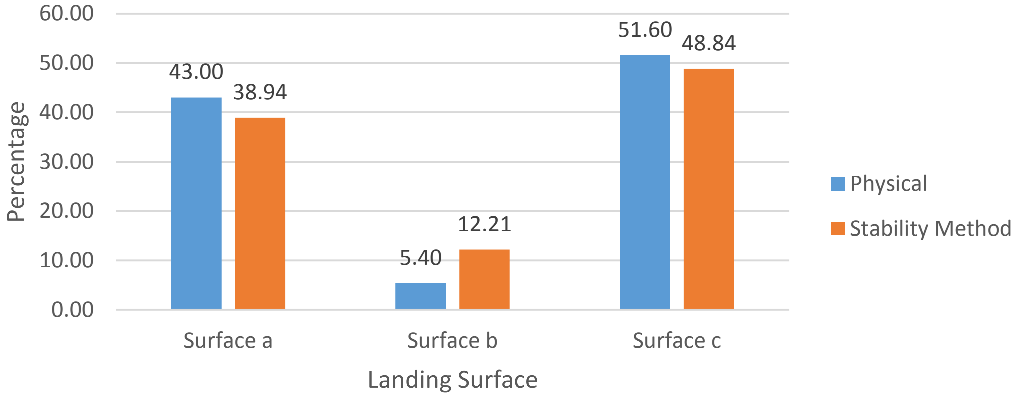


Fig. 7. Results comparison between physical testing data and original stability method data for rectangular prism with aspect ratio 0.56

The application of the chi-square test gives a p-value of 0.2234 and chi-square test value of 4.3774. Thus, it can be said with 90% confidence that the difference in percentages shown is due to chance.

b. Original Stability Method Application to Object 1

Object 1 is a plastic cap that is used to investigate the application of the theory to objects with curved surfaces and holes. Fig. 8 shows the SolidWorks drawing of the part, which has three unique landing surfaces, all of which are highlighted in Fig. 9. The surface area and its distance from the center of gravity are computed using the measurement tool in SolidWorks and are detailed in Table 6. Values of surface area and distance from the surface from the center of gravity of Object 1 for Original Stability Method can be found in Appendix B.

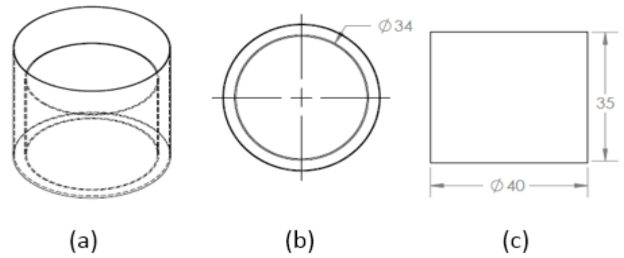


Fig. 8. Object 1, a part used to investigate the limitations of the Stability Method: (a) Isometric view of Object 1 with hidden lines; (b) Top view with dimensions, measured in millimeters; (c) Side view with dimensions measured in millimeters.

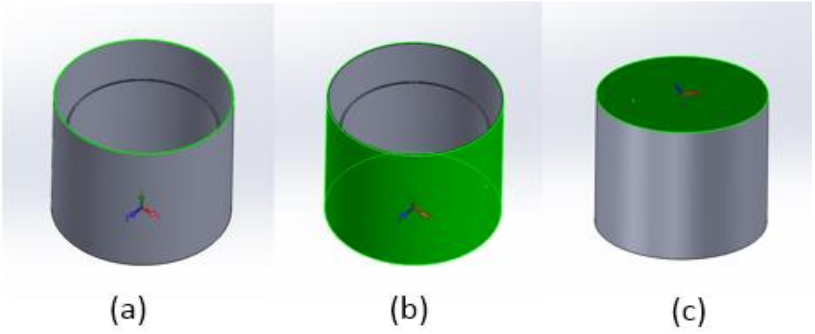


Fig. 9. Different landing surfaces of Object 1 considered in the Original Stability Method, area highlighted in green: (a) Landing Surface 1; (b) Landing surface 2; (b) Landing surface 3.

Fig. 10 is a histogram plot comparing the two sets of data. The p-value is less than 0.00001 and the chi-square test value is 321.7574. From this, it can be said with 90% confidence that the difference in the two sets of data are significant and are 90% likely not due to chance. Thus, the application of the Stability Method to Object 1 is considered inaccurate.

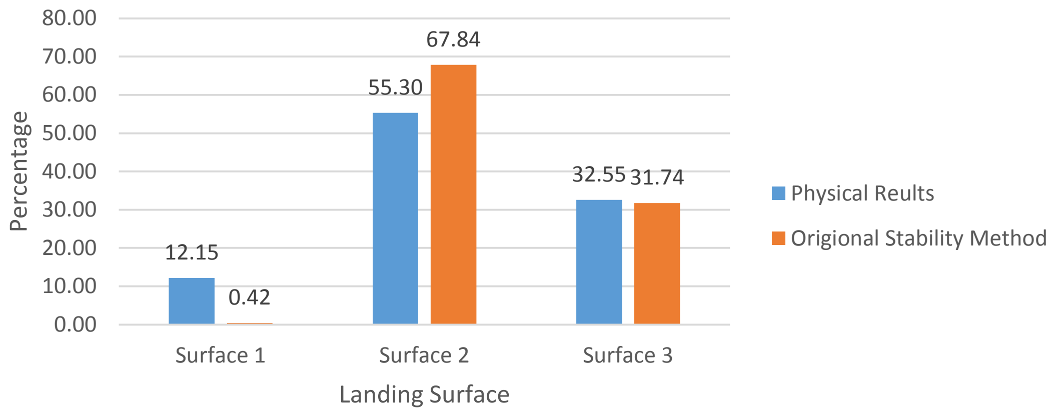


Fig. 10. Results comparison between physical testing results and stability method results for Object 1

c. Original Stability Method Application to Object 2

The theory was applied to Object 2 shown in Fig. 11, a part with a more complex shape, more specifically with landing surfaces that are curved where the area of the landing surface is not as straight forward as the previous example. Object 2 is made out of wood and has five different stable orientations, all of which are shown in Fig. 12.

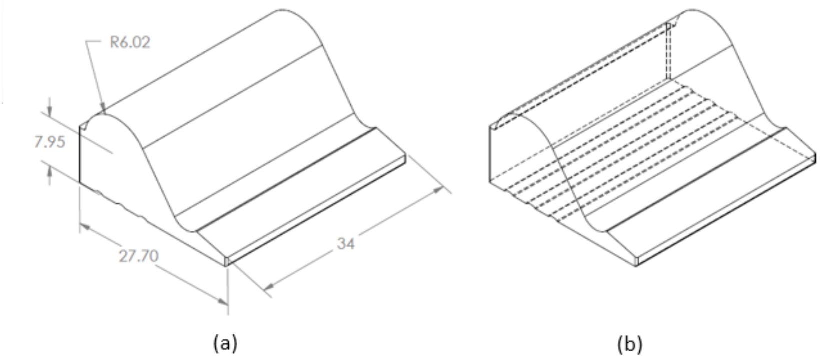


Fig. 11. Object 2, a Complex Part used to investigate the limitations of the Stability Method. Dimensions measured in millimeters: (a) Isometric view of Object 1 with dimensions, measured in millimeters; (b) Isometric view of Object 1 with hidden lines.

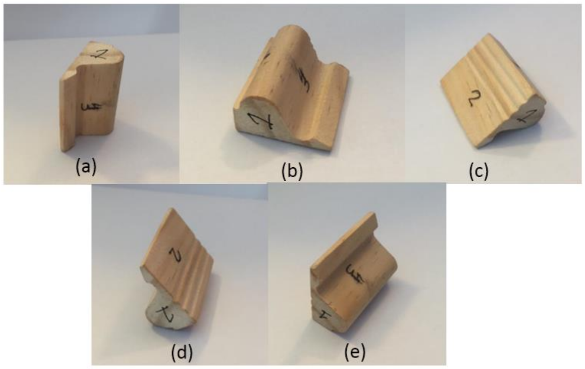


Fig. 12. Different stable orientations of object 2; (a) Stable Orientation on Landing Surface 1; (b) Stable Orientation on Landing Surface 2; (c) Stable Orientation on Landing Surface 3; (d) Stable Orientation on (d) Landing Surface 4; (e) Stable Orientation on Landing Surface 5

The comparison of the results obtained from physical testing and the Original Stability Method is shown in Fig. 13. The application of the chi-square test comparing the two sets of data yields a p-value of less than 0.00001 and a chi-square test value of 536.8054. Thus, it can be said with 90% confidence that the difference in values of the two data sets is not due to chance.

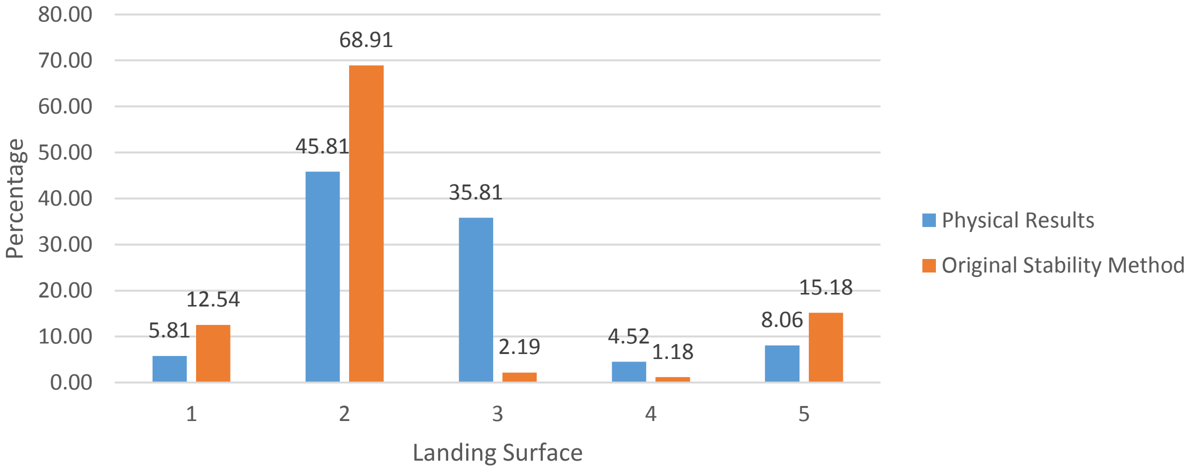


Fig. 13. Results comparison between physical testing results and stability method results for Object 2

d. Original Stability Method Application to Product

A simplified model of the part of interest was modeled in SolidWorks to be used for analysis. The model consists of all the key features, dimensions and the general shape of the part of interest. The model is made of a uniform material. Fig. 14 is an isometric view of the simplified model of the part.

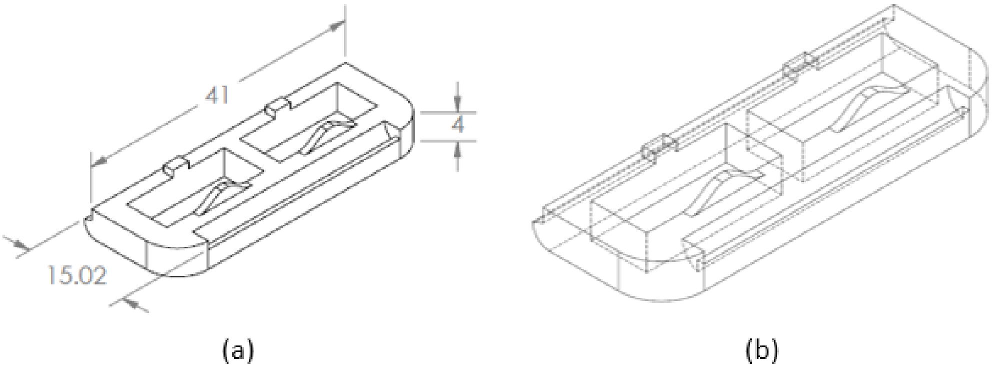


Fig. 14. Isometric View of the Simplified Part used for analysis: (a) Isometric view with dimensions, measured in millimeters; (b) Isometric view with hidden lines.

The Stability Method was applied to the model, which has two different landing surfaces. The surface areas considered according to the original theory is shown in Fig. 15 and the distance of the surface from the center of gravity is calculated using the measurement tool in SolidWorks. The values of the total surface area of the landing surfaces and their distance from the center of gravity are shown in Table 12.

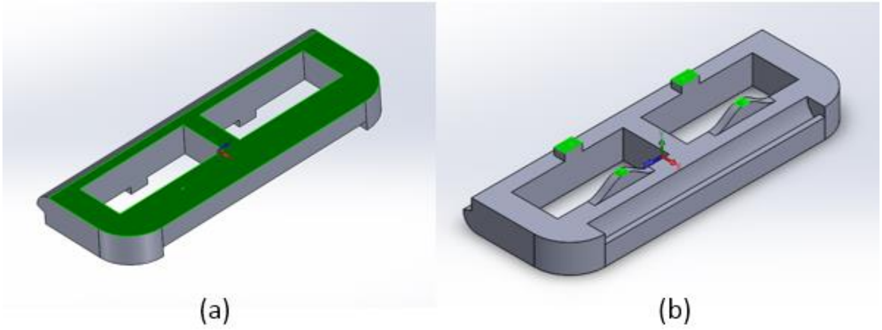


Fig. 15. Application of Original Stability Method to the simplified product, surface area considered highlighted in green: (a) Surface area considered for Landing Surface 1; (b) Surface area considered for Landing Surface 2.

The comparison between the data obtained from physical testing and the stability method is shown in Fig. 16. The p-value of the two data sets is less than 0.00001 and the chi-square test value is 849.8981, which means that the difference in the two sets of data is significant and is not due to chance.

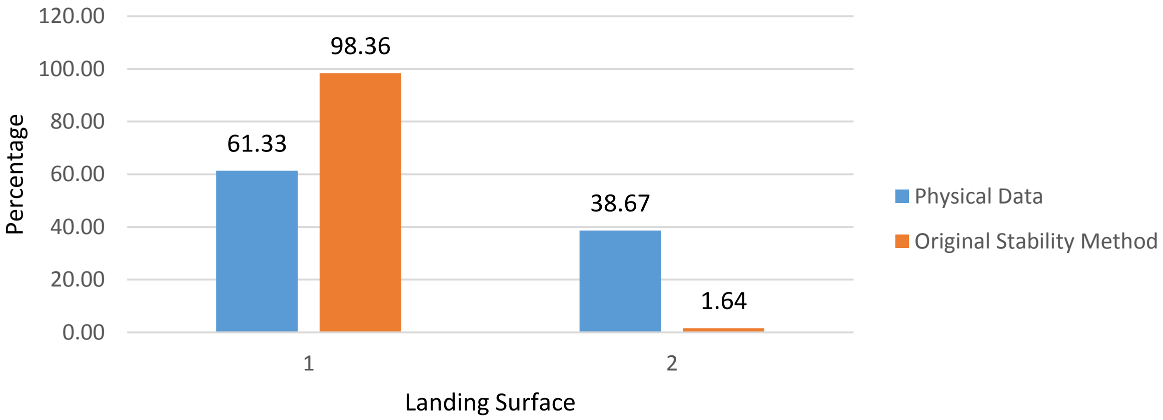


Fig. 16. Results comparison between physical testing results and stability method results for application to Simplified model of the product

From evaluating the results from Object 1, Object 2 and the simplified product and the application of the Stability Method, it can be said that the method works with non-complex parts and the theory needs to be modified in order to obtain accurate results when applied to models with complex geometries. Thus, a Modified Stability Method was created. The method deals with taking into account complex parts with holes and curved surfaces.

3.2.4. Modified Stability Method

a. How it works

The Modified Stability Method was primarily developed to take into account more complex geometrical features, more specifically features involving curved surfaces and holes. The surface area that was considered as part of the equation in the calculations was based on the area of the landing surface that was directly in contact with the ground, which works with simple objects, as demonstrated by the rectangular prim example in Section 3.2.3. However, as the results from Object 1 and Object 2 demonstrated, it does not work with more complex geometrical features.

The Modified Stability Method takes into consideration the surface area that is not in direct contact with the ground and considers the whole plane that is created by the surface in contact in the calculation for the surface area. y_i is the distance between the plane created from the surfaces that are directly in contact with the ground and the center of gravity, the same as the original method and can be measured using the measurement tool in SolidWorks.

The Modified Stability Method was applied to Object 1 and Object 2 so that the method can be validated and analyzed, then the method was applied to the simplified model of the product.

b. Modified Stability Method Application to Object 1

The surface areas considered for the three different landing surfaces for Object 1 is shown in Fig. 17. The surface area that was considered for Landing Surface 1 is changed from only considering the surface area that is directly in contact with the ground, which is the thin outer rim of the cap to considering surface area of the whole plane created by the outer rim. The surface area that is considered for Landing Surface 2 and 3 remains unchanged.

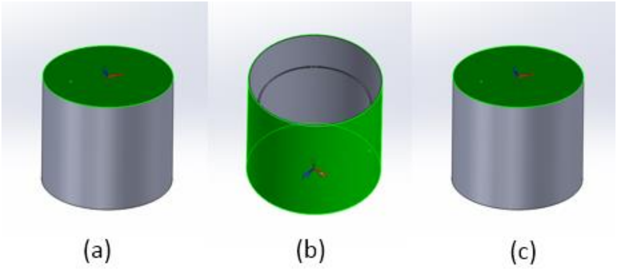


Fig. 17. Different landing surfaces of Object 1 considered in the Modified Stability Method, area highlighted in green:
(a) Landing Surface 1; (b) Landing surface 2; (c) Landing surface 3.

The effect on the change in surface area considerations can be seen in Fig. 18, which shows a histogram comparing the results obtained from physical testing, the Original Stability method and the Modified Stability Method for Object 1. The data obtained from the Modified Stability Method was compared to the data from the physical testing using the chi-square test, giving a p-value of 0.4247 and chi-square test value of 1.6661. Thus, it can be said with 90% confidence that the difference in the data obtained from physical testing and the Modified Stability Method is not significant and is most likely due to chance.

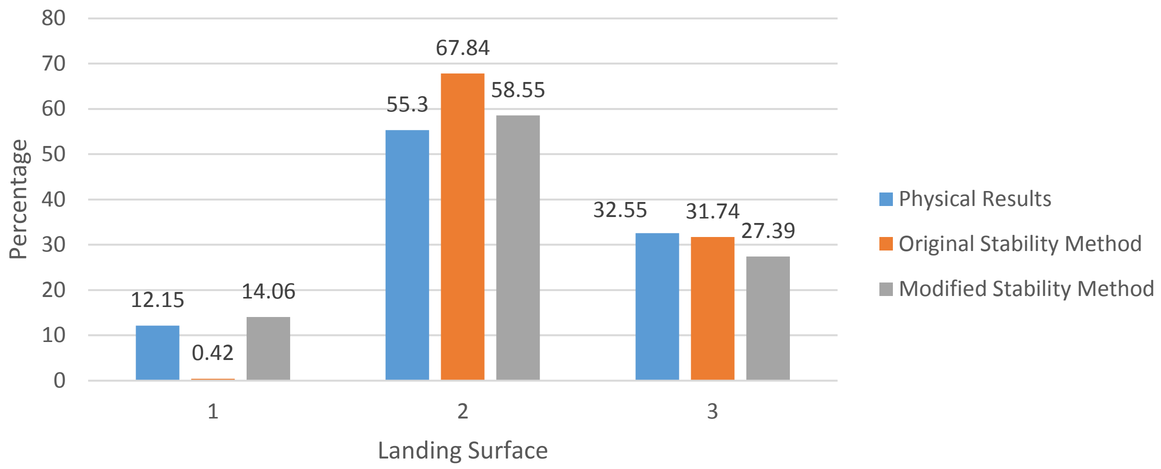


Fig. 18. Results comparison between physical testing results and original stability method and modified stability method results for Object 1

c. *Modified Stability Method Application to Object 2*

The different surface areas considered in the Modified Stability Method calculations for Object 2 are shown in Fig. 19. The surface area that was considered for Landing Surfaces 1, 2 and 4 remains unchanged since they are all flat surfaces. The two landing surfaces that were affected by the Modified Stability Method are Landing Surfaces 3 and 4. As shown on Fig. 19, instead of considering only the areas that are directly in contact with the ground for Landing Surface 3, the whole plane that is created from the two sides that are directly in contact with the ground is now considered part of the surface area calculation. The same applies to Landing Surface 4. This increases the surface area of each of the landing surfaces considerably and thus effected the distribution of the results.

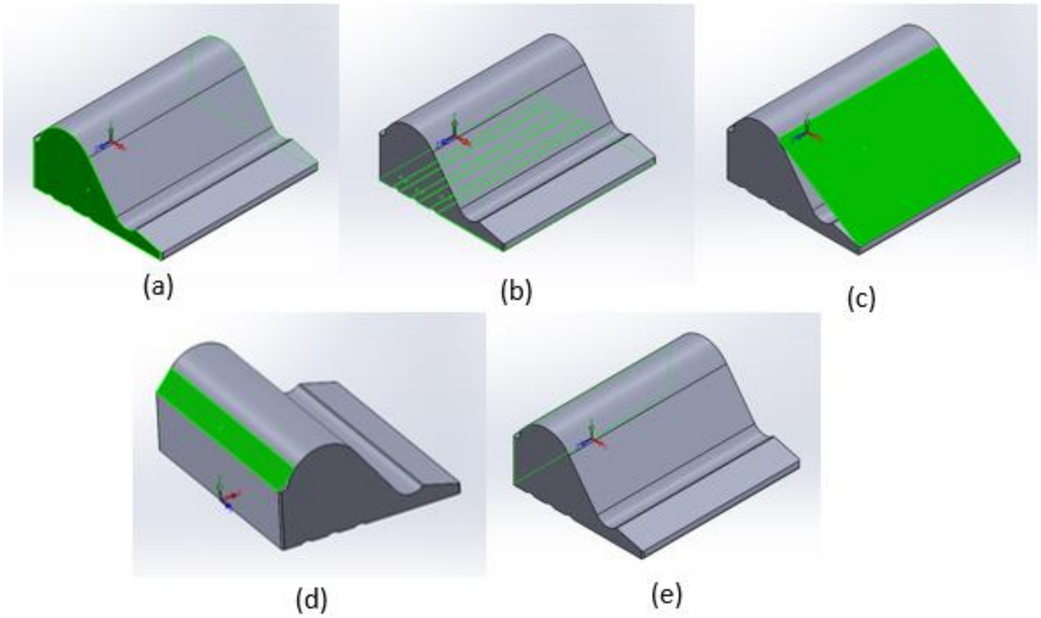


Fig. 19. Different landing surfaces of Object 1 considered in the Modified Stability Method, area highlighted in green: (a) Landing Surface 1; (b) Landing surface 2; (c) Landing surface 3; (d) Landing Surface 4; (e) Landing Surface 5.

The distribution of the percentage occurrence for Object 2 is shown in Fig. 20. The percentage occurrences from the Modified Stability Method was compared to the data obtained from physical testing to give a p-value of 0.8507 and a chi-square test value of 1.3288, which can be found in Table 17. Chi Square Test Results where Physical Testing = PT, Original Stability Method = OSM, Modified Stability Method = MSM, New Modified Stability Method = NMSM. This means that it can be said with 90% confidence that the differences in values of the two data sets are not significant and due to chance.

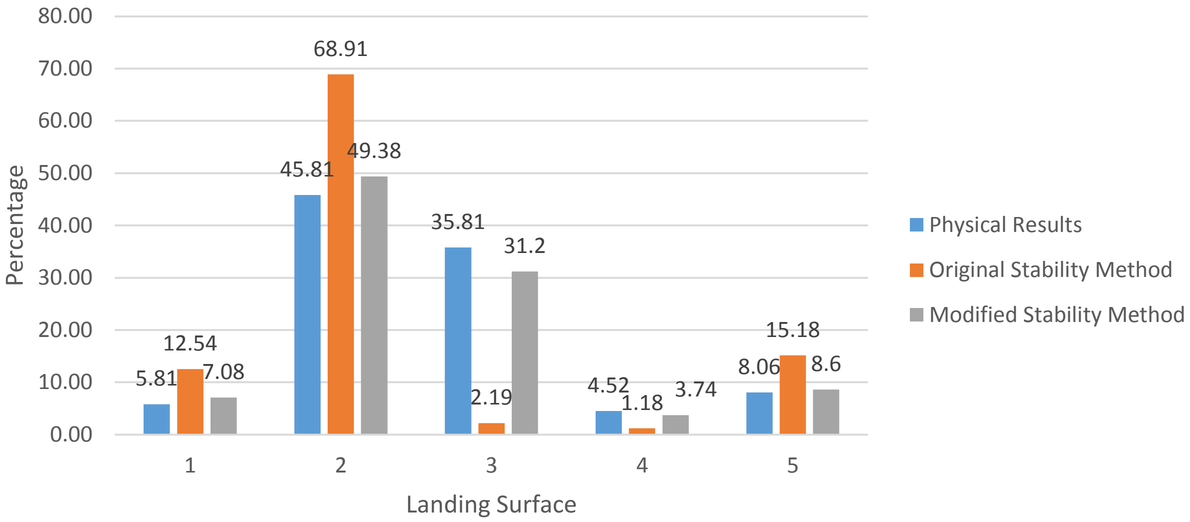


Fig. 20. Results comparison between physical testing results, Original Stability Method and Modified Stability Method results for Object 2

d. Modified Stability Method Application to Product

The Modified Stability Method was re-applied to the simplified model of the product. The new method affects both the areas considered for landing surfaces 1 and 2. This is illustrated in Fig. 21.

According to the Modified Stability method, the area considered for Landing Surface 1 is the area that is directly in contact with the ground in addition to the area of the two extrusions in the center of the model, making the plane as shown. The area that is considered for Landing Surface 2 is the area of the plane created from the four surfaces that are directly in contact with the ground, making the plane as shown.

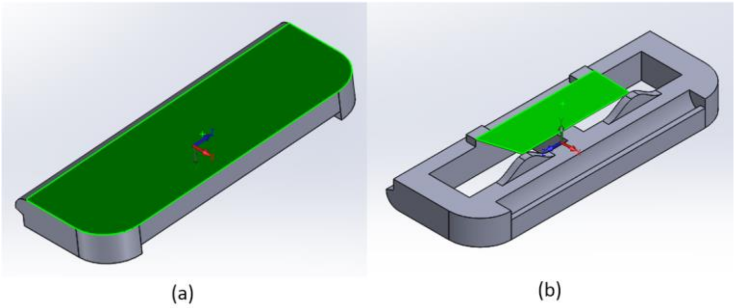


Fig. 21. Different landing surfaces of simplified product considered in the Modified Stability Method, area considered highlighted in green: (a) Landing Surface 1; (b) Landing surface 2.

The data obtained from applying the Modified Stability Method to simplified product is compared to the data collected from physical testing, shown in Fig. 22. The p-value is less than 0.00001 and the Chi-square test value is 179.9943, making the numerical difference in the two sets of data significant with 90% confidence.

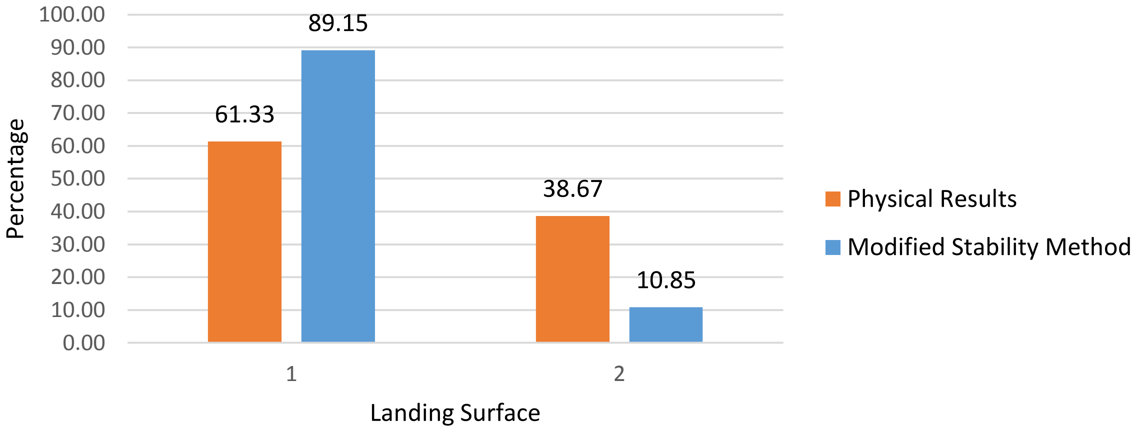


Fig. 22. Results comparison between physical testing and Modified Stability Method results for simplified product

The Modified Stability Method gives accurate results for simple objects with curved surfaces and extrusions, as illustrated by its application to Object 1 and 2. The method brings the percentage distribution between the two landing surfaces of the simplified product closer to the data obtained from physical testing. However, it is evident that there are other aspects of the method that need to be modified, in order for application to parts with complex geometrical features such as the product.

3.2.5. New Modified Stability Method

a. How it works

This method takes into account the surfaces that an object can land on, but cannot stabilize on. The surface areas of such surfaces are added onto the surface area of the landing surface that the object stabilizes on after landing on the unstable surface. The unstable surface the object can land on comes in two forms: the surface can be a plain flat surface or less obvious landing surfaces created by the complex geometry of the shape in question. An example of the first form is illustrated by the application of the New Modified Stability Method to Objects 3 and 4 and the latter is illustrated by the application of the method to the simplified model of the product.

Two different approaches were considered; the first approach only takes into consideration the landing surface‘s distance from the center of gravity for the stable surface. The surface area of the unstable surfaces is then added to the surface area of the stable landing surface, and then calculated based on the Stability Method equation. The second approach involves the consideration of the unstable landing surface’s distance from the center of gravity. For the first approach, all the different unstable surfaces are considered as individual landing surfaces and calculated as if the landing surfaces are stable. Once the percentage distribution is found, the percentage of occurrence from the unstable surfaces is summed up with the percentage occurrence of the surface in which it stabilizes on. The sum of the stable and the unstable surfaces are considered the total percentage occurrence of the particular landing surface

The New Modified Stability Method was applied to the simplified model of the product and then to Objects 3 and 4 to validate that the application of the theory works with more than one object.

b. New Modified Stability Method Application to Product

The New Modified Stability Method is applied to the simplified model of the product as shown in Fig. 23. The surface areas considered for Landing Surface 1 remains unchanged from the areas considered in the Modified Stability Method. The most drastic change is with Landing Surface 2, where four more different surface areas are added. The surface areas are all areas of planes that are created from the model landing at an angle and stabilizing on Landing Surface 2. A more detailed explanation about the different surface areas considered as part of the New Modified Stability Method for the simplified model of the product can be found in Appendix F.

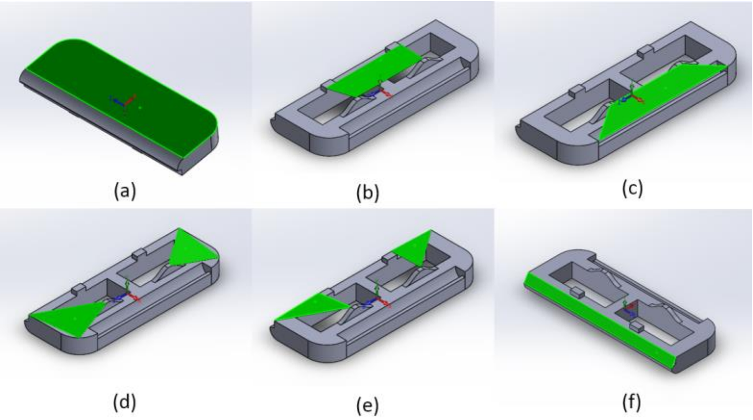


Fig. 23. Different landing surfaces of simplified product considered in the New Modified Stability Method, area considered highlighted in green: (a) Landing Surface 1; (b) Landing surface 2; (c) Landing surface 2.1; (d) Landing Surface 2.2; (e) Landing Surface 2.3; and (f) Landing Surface 2.4.

The results obtained from the two approaches are shown compared to the data obtained from physical testing in Fig.24. The percentage error calculation conducted on the two methods used the data from the physical testing as the expected result and the data obtained from the two approaches as the theoretical results. The average percentage error for Approach 1 was 5.10% while Approach 2 was 34.11%. The calculation of these results can be found in Appendix G. The application of Chi-square test found in Appendix E. Table 17, gives a p-value of 0.6151 and chi-square test value of 0.2527 for Approach 1 and a p-value of 0.6152 and chi-square test value of 4.2659 for Approach 2. From this, it can be said with 90% confidence that the difference in the data set obtained from Approach 1 and physical testing is due to chance and it is not statistically significant with the given confidence interval. On the other hand it can be said with 90% confidence that the difference in the data set obtained from Approach 2 and physical testing is 90% likely not due to chance. Thus, it can be concluded that Approach 1 is more accurate.

Approach 2 of the New Modified Stability method was further validated by applying the approach to two other parts that consisted of unstable landing surfaces. These results were used to analyze the accuracy of the theory verses the results obtained from physical testing using the chi-square test.

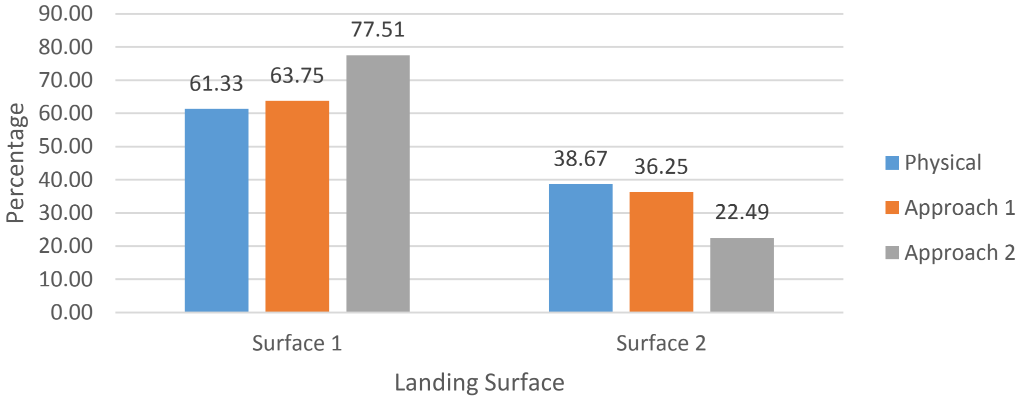


Fig. 24. Results comparison among physical testing, original stability method and modified stability method for the simplified product

c. *New Modified Stability Method Application to Object 3*

Object 3 was used prove that the New Modified Stability Method works with parts that do not resemble the product in which the theory was developed from. The isometric view and dimensions of the part is shown in Fig. 25.

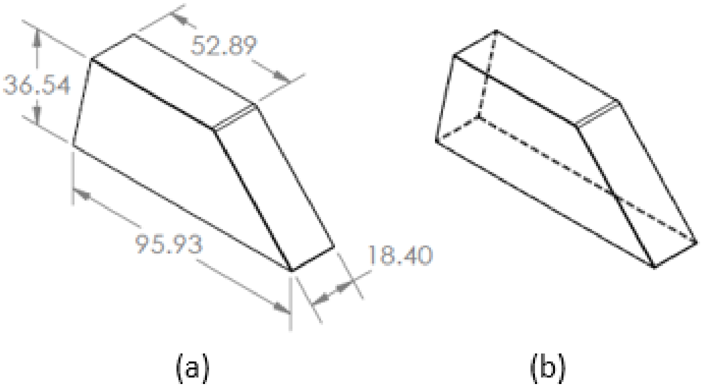


Fig. 25. Object 3, a part used to prove the concept of the New Modified Theory: (a) Isometric view of Object 3 with dimensions, measured in millimeters; (b) Isometric view of Object 3 with hidden lines;

The part consists of four stable landing surfaces and one unstable landing surface. The values of the surface areas of the landing surfaces and its distance from the center of gravity calculated using SolidWorks can be found in

Table 10. The different landing surfaces of Object 3 are illustrated in Fig. 26. Landing Surface 3.1 shown in Fig. 26 is an unstable landing surface, if the part hits the ground at Landing Surface 3.1, it will stabilize on Surface 3. Following Approach 1 for the percentage occurrence calculation, the surface area for Landing Surface 3.1 is summed with the surface area for Landing Surface 3 and calculated in the Stability Method equation as a unified surface.

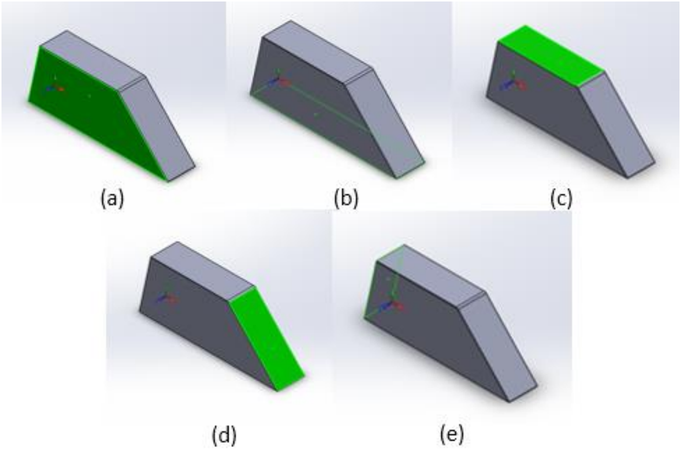


Fig. 26. Different landing surfaces of Object 3 considered in the Modified Stability Method, area highlighted in green (a) Landing Surface 1; (b) Landing surface 2; (c) Landing surface 3; (d) Landing Surface 3.1; (e) Landing Surface 4.

The data obtained from Approach 1 of the New Modified Stability Method is compared to the data obtained from the physical results in Fig. 27. The application of the Chi-square test gives the p-value of 0.9220 and a chi-square test value of 0.5773. Thus, it can be said with 90% confidence that the differences in the two sets of data are likely due to chance.

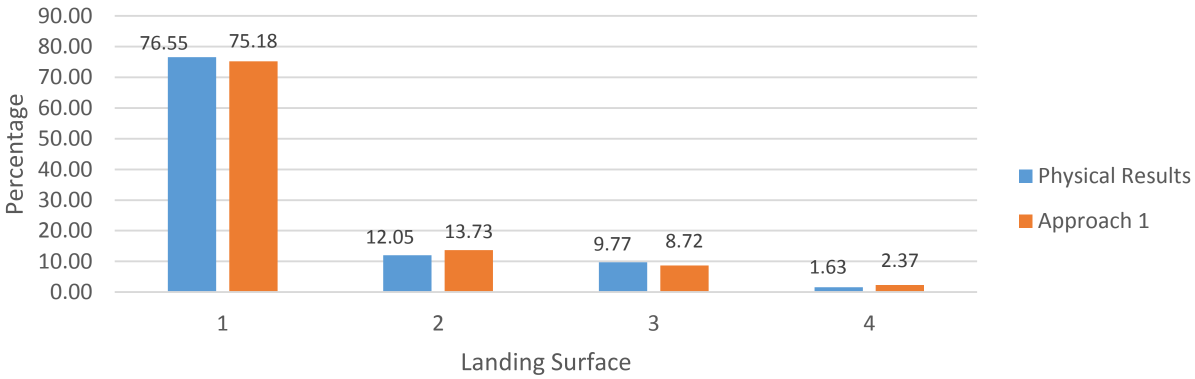


Fig. 27. Results comparison between physical testing and Approach 1 of the New Modified Stability Method for Object 3

d. New Modified Stability Method Application to Object 4

The New Modified Stability Method application to Object 4 can be found in Appendix H. Application of New Modified Stability Method to Object 4

3.2.6. Recommendations to Alter Designs for the Product

The design of a part can be altered to change the ratio of the landing orientations. The Stability Method equation takes into account the surface area of the landing surfaces and their distance from the center of gravity. Thus, the ratio of the landing orientations can be achieved in two ways: by changing the surface area of the landing surfaces or by changing the location of the center of gravity. Both these methods involve changing either the dimensions, adding protrusions and extrusions or by changing the material distribution and properties.

The main focus of this project is to provide a tool that will be able to predict the natural part orientation of the product of interest. Thus, this section will focus on the mathematical processes involved in providing recommendations to change the ratio of the landing orientations for the product of interest.

For calculation simplification purposes, the product, which has two stable resting orientations and was simplified into a cuboid with landing surface areas h_1 and h_2 as their respective distances from the center of gravity, as illustrated in Fig. 28.

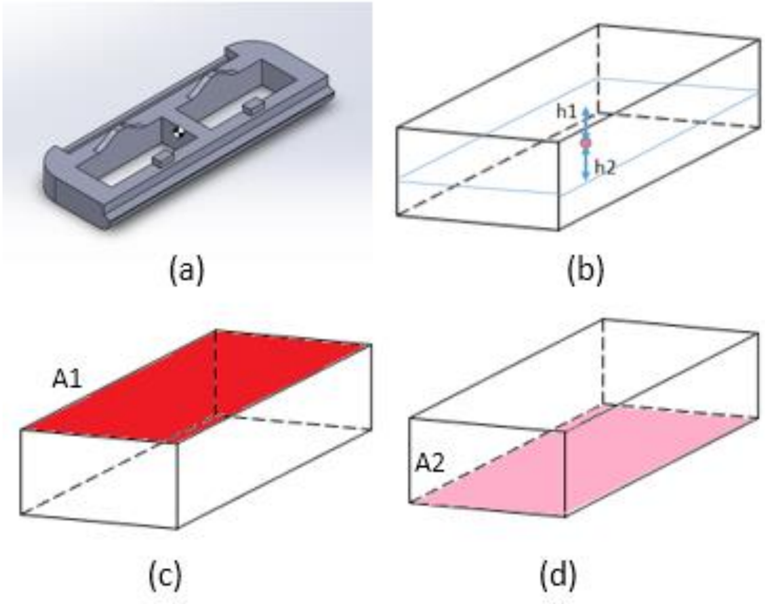


Fig. 28. Simplification of the product into a cuboid for calculation purposes; (a) Simplified product model; (b) The simplified product model shown as a cuboid with the distances from the center of gravity, h_1 and h_2; (c) Landing Surface area A_1; (d) Landing Surface area, A_2.

The Stability Method Equation can be written in terms of the variables expressed above:

$$p_1 = \frac{\frac{A_1}{h_1}}{\frac{A_1}{h_1}+\frac{A_2}{h_2}} \tag{5}$$

$$p_2 = \frac{\frac{A_2}{h_2}}{\frac{A_1}{h_1}+\frac{A_2}{h_2}} \tag{6}$$

Recommendations regarding what the location of the landing surface's distance from the center of gravity should be for the desired orientation can be made by assuming that there is no change in the surface area of the landing surfaces when there is a change in the location of the center of gravity. Thus, this creates a simultaneous equation where h_1 and h_2 can be solved using any mathematical computation program.

The application of the equations above to the simplified model of the product using Approach 1 of the New Modified Stability Method with the desired percentage distribution p_1 and p_2 of 0.54 and 0.46 gives the new values of h_1 and h_2, 2.473 and 3.027 respectively. The computation was conducted using MATHCAD and can be found in Appendix I.

The CAD model of the simplified product was modified by creating protrusions on the model which would not affect the landing surfaces to give the closest value of h_1 and h_2 as much as possible, with the values 2.49 and 3.02. An isometric view of the modified model is shown in Fig. 29. The model was three-dimensionally printed for physical testing. The results from the physical testing are compared to the expected theoretical values in Fig. 30. It can be said with 90% confidence that the difference between the two sets of data are likely due to chance, with the p-value of 0.8060 and Chi-square test value of 0.06031.

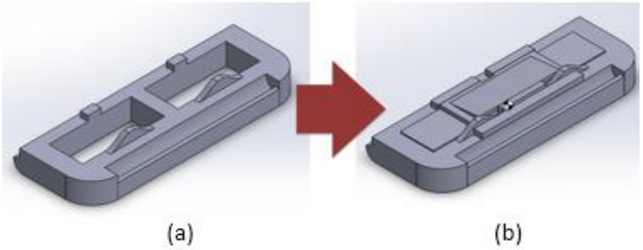


Fig. 29. Modifications to the simplified product; (a) Original Model; (b) Modified Model

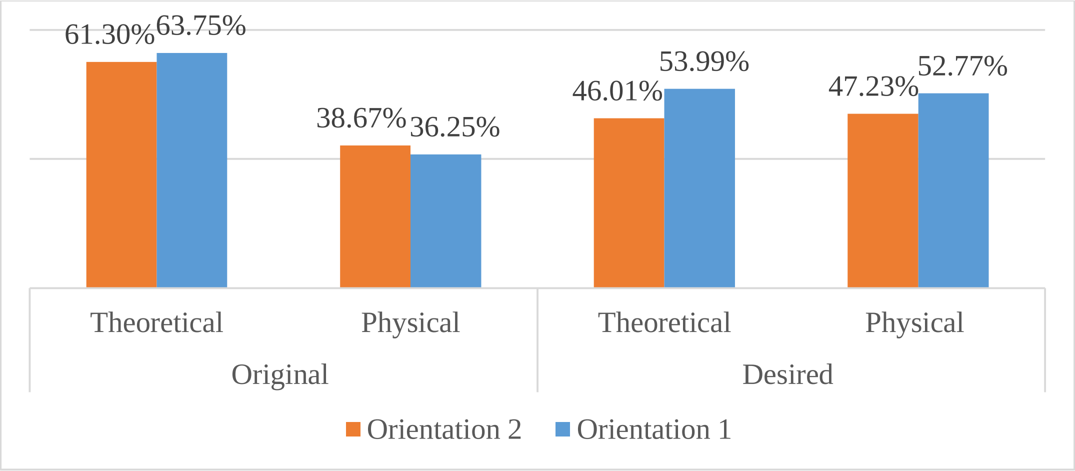


Fig. 30. Results comparison between physical testing and Approach 1 of the New Modified Stability Method for modified simplified product compared to the original simplified product

3.3. Computational Method

3.3.1. Finite Element Analysis (FEA)

a. Software

ABAQUS was used to conduct the FEA analysis required for the project. The main reasons for this were that it was the sponsor's choice of software and it is able to fulfil all the software requirements that the project required.

b. How FEA Works

FEA is a method that is used to solve physical world scenarios. The premise behind FEA is breaking a system down into nodes, which then are used to create elements. These elements provide the structure and a method of connecting the nodes together. The nodes are points in which calculations are based from. The more nodes in a system, the smaller the elements will be, and in return the more accurate the simulation results. Once the nodes and elements are defined in the simulation, the FEA software is then able to calculate the given variables and conditions of each node over a time frame. The time frame is called a step or iteration. In most cases, with exception of highly linear problems, the smaller the irritation becomes, the more accurate the results are. This phenomenon can also be observed in mathematical integration, the smaller the change in time is, and the more closely the integration will fit the curve. When the step size becomes infinitesimally small the results will almost perfectly fit the desired curve. The step sizes in FEA are determined through different methods outlined in Section 3.3.1.b.i.

i. *Explicit Vs. Implicit Dynamics*

In ABAQUS, two main approaches can be used to evaluate a simulation implicitly and explicitly. The implicit method is simpler from a computational stand point, and therefore the method works better with static and linear problems. When using the implicit method for non-linear simulations ABAQUS uses linear approximations during each time interval. The less linear the simulation is, the smaller these time steps will be in order to achieve convergence of the system. This becomes increasing difficult with non-linear simulations. Although when a simulation is purely linear, the time steps of the system are able to increase in size to whatever size is requested by the user, allowing for faster simulation results. Thus, this method only has the capability to solve simple contact problems. This is due to the none-linearity of these problems and difficulty of calculating impact forces. Along with these limitations the implicit method is often extremely redundant, since it often has to solve the same equations multiple times to check for convergence during each iteration. Moreover, when modeling larger models with a higher node count the implicit method will take longer than the explicit method with regards to run time. This is because its runtime scales exponentially with the node count while the explicit method scales near linearly. So the implicit method is best used when there are relatively small models in linear and static simulations.

The second way to evaluate a simulation is using the explicit method. This method is used for non-linear simulations where wave propagation is prevalent, and is the method for the simulations involved in this project. Unlike the implicit function the explicit function does not have to solve for convergence of the function at each iteration. Instead the explicit function determines the step size by the equation:

$$\Delta t \leq \frac{2}{\omega_{max}}(\sqrt{1+\varepsilon_{max}^{2}-\varepsilon_{max}}, \tag{7}$$

where ω_{max} is the largest frequency in the equation, and ε_{max} is the maximum fraction of critical dampening in the system. This allows the simulation to move from step to step without having to re-evaluate each step and check for convergence. In addition to this the stable time increment size can also be estimated by:

$$\Delta t = \frac{L_{\min}}{\frac{\sqrt{\frac{E*v}{(1+v)*(1-2v)}+2*(\frac{E}{2*(1+v)})}}{\rho}}, \tag{8}$$

where $L_{\min}$ is the smallest element dimension in the mesh, *E* is the materials young's modulus, *v* is the materials Poisson's ratio, and ρ is the materials density. In addition to this the bottom term in Equation 8 represents the dilatational wave speed; which is how fast a wave propagates through the material. From these equations a rough estimate of how long the simulation will take with the number of required iterations is:

$$n \approx T_{max} * \left(\frac{1}{L_c}\right) * \sqrt{\frac{\frac{E*v}{(1+v)*(1-2v)}+2*(\frac{E}{2*(1+v)})}{\rho}} \ , \tag{9}$$

where T_{max} is the total simulation time, and L_c is the characteristic length associated with an element (Dassault Systems Simulia Corp, 2012).

Equation 9 shows that refining the mesh will increase the simulations run time, however the amount of increase is dependent on the mesh type. While increasing the density whether manually or from ABAQUS' mass scaling feature will decrease the step time and simulation run time. This is because increasing the density of the part also decreases the propagation wave speed across the part. Once the step size is set ABAQUS then uses the central difference rule to integrate the equations of motion explicitly through time. At the beginning of the increment the program solves for the dynamic equilibrium:

$$M\ddot{u} = P - I, \tag{10}$$

where M is the nodal mass matrix, $\ddot{u}$ is the nodal accelerations, P is the externally applied forces and I is the internal forces (Dassault Systems Simulia Corp, 2012). The acceleration at the beginning of the current time increment is then calculated using the equation:

$$\ddot{u} = M^{-1} * (P - I)\ . \tag{11}$$

Once ABAQUS has the acceleration it is then integrated through time in order to find the velocity of each node. This is done using the equation:

$$\dot{u}_{\left(t+\frac{\Delta t}{2}\right)} = \dot{u}_{\left(t+\frac{\Delta t}{2}\right)} + \frac{\Delta t_{(t+\Delta t)}+\Delta t_{(t)}}{2}\ \ddot{u}_t\ . \tag{12}$$

After the acceleration is integrated through time ABAQUS does the same with the velocity in order to find the displacement over the time increment (Dassault Systems Simulia Corp, 2012). In addition to the equations of motions ABAQUS also uses a conservation of energy of:

$$E_I + E_V + E_{FD} + E_{KE} + E_W + E_{PW} + E_{CW} = E_{Total} = constant, \quad (13)$$

where E_I is the Internal energy, E_V is the viscous energy dissipation, E_{FD} is the work done by friction, E_{KE} is the kinetic energy, E_W is the work done by externally applied loads, E_{PW} is the work done by contact penalties, and E_{CW} is work done by contact constraints (Dassault Systems Simulia Corp, 2012). E_1 can be further broken up into its components:

$$E_1 = E_E + E_P + E_{CD} - E_A\ , \quad (14)$$

where E_E is the recoverable strain energy, E_P is the energy dissipated through inelastic processes (zero in our case), E_{CD} is the energy dissipated though viscoelasticity or creep, and finally E_A is the artificial strain energy (Dassault Systems Simulia Corp, 2012). Equation 13 is generally balanced within 1% accuracy for each individual step. This equation is used in order to make sure the transfer of energy between nodes and constraints are always constant.

The explicit method is best used for non-linear simulations, as it can achieve much faster run times when compared to the implicit method. The implicit method is also the main method used when performing drop tests due to its more complex contact analysis procedures.

c. FEA Analysis Walkthrough

i. *Importing Parts*

In order to start an analysis of a part, the parts geometric properties are needed. These can be imported in the form of a STEP file from SolidWorks. However, any 3D modeling software can be used for this step as long as it is able to output a STEP file. The parts can also additionally be modeled inside of ABAQUS's GUI as an alternative method. After the parts are modeled in ABAQUS using ether of the above methods the simulation can be set-up.

ii. *Assigning Material Properties*

One of the benefits of using FEA is that the material conditions of the parts are taken into account. Since the material properties are taken into account they must also be entered into ABAQUS. For the simulations at hand two materials were defined. One material was for the part that is being dropped, while the second material is used in order to define the properties of the ground. The properties that defined the materials are the Young's Modulus, density and Poisson's ratio. The yield stresses of the materials were not added. This was because the simulation is under the assumption that no plastic deformation would occur. Since no plastic deformation would occur the yield stress would not have to be calculated as this would only add to the computation time of the simulation. In order to assign the materials to their correct part, sections must be made. If the part only has a single material the whole model must be selected, if the part has multiple materials custom sections must be made in order to properly define where each material starts and ends.

iii. *Meshing of the Model*

The next step in the process of setting up the simulation is meshing the parts. In FEA one of the most important processes, which ensures the accuracy of the final results is the meshing of the model. A model's mesh is made out of interconnected nodes, these nodes are points in which forces and variables can be calculated on. Nodes can be thought of almost like pixels in a picture; with more pixels a clearer picture will be produced. However unlike pictures, nodes accompany a space in 3D, so with a greater amount will produce a more defined model. This is shown in Fig. 31

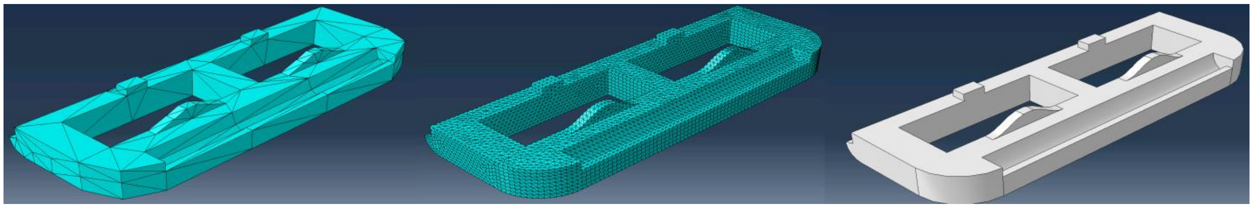


Fig. 31: Mesh size comparison, an example of how increased node counts lead to a more defined model

Fig. 31 shows how the model becomes more defined as the node count in the mesh increases, this especially seen on curves and more complicated geometry. The increased definitions not only allows for visual changes, but it also allows the FEA software to calculate more data points and get a more accurate result of what is happening in the simulation. However since the software is undergoing more calculation the runtime of said simulations will increase with the amount of nodes in the system. This creates a predicament where enough nodes are wanted to give an accurate result but, runtime of the solution also needs to be as low as possible. In order to calculate the correct amount of nodes, also known as the mesh size, a convergence test must be conducted. A convergence test is done by slowly decreasing the mesh size until the results of the simulation no longer changes by an acceptable amount; a graph of this is shown in Fig. 32.

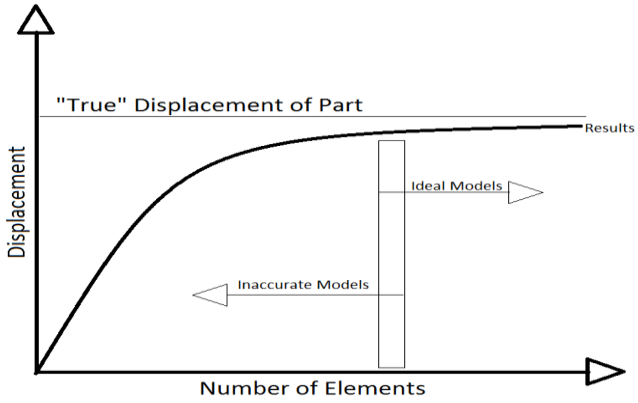


Fig. 32. Graphic view of typical converge tests

Figure 32 shows that once the mesh size gets to a certain point the changes in the results become insignificant compared to the run time increase that is caused from this. The criteria for measuring the convergence will also change depending on what is being measured, for example stresses tend to converge much faster than displacements. Typically convergence tests are done against the most important data that is being obtained from the simulation. For example if a typical impact test is being done, the convergence test would test to see when the stresses converge. For this project the most important output is the surface in which the part lands on. Because of this when the convergence tests were done, they were compared based on which surface they landed on. These convergence tests are later discussed in Section 3.3.1.ix. As for the initial set up of the simulation the mesh size doesn’t have a large impact since before any data should be taken, convergence of the mesh would first be assessed. Because of this for now use the default seed-size that is provided by ABAQUS.

iv. *Assigning Part Surfaces*

When dealing with contact modeling one must first decided on how the contact will be defined. For this project the best form of contact interaction was surface to surface, as it takes less computation power than full contact models. In addition to this it also provides surface contact forces, which can be cross-referenced to find the landing surface, which is later explained in Section 3.3.1.viii. Since surface-to-surface contact was being used the contacting surfaced have to be defined. This is done inside the part model by creating a surface for each individual face that has the potential to come into contact with the ground.

v. *Assembling the Model*

In ABAQUS the assembly is an instance where meshed parts may be imported into, in order to partake in a simulation. So once the creation of the individual parts is finished, an assembly instance should be formed to contain these created parts. The creation of the assembly is carried out by importing each individual part into the instance that was just defined. The parts are then translated and rotated into the proper orientation in preparation for the simulation. The translations and rotations are done though vector commands, these can be either from manually select points in order to make up the vectors, or from using numerical values in order to obtain the vectors needed. A rough configuration of how this should look us shown in Fig. 33.

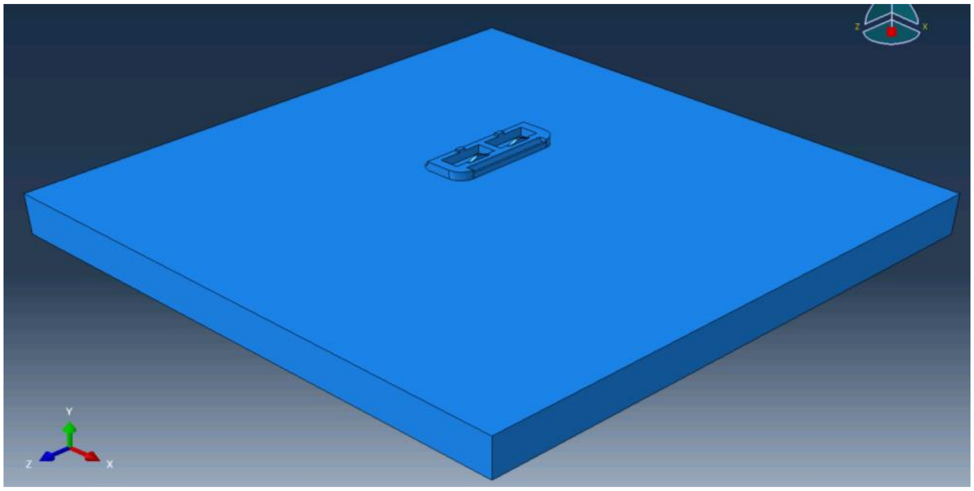


Fig. 33. Assembly view of basic drop simulation.

It should be noted that the part’s distance from the ground surface should equal approximately 5% more than the parts longest side. This is done so when the script randomizes the initial orientation it will not pass into the ground surface, and create errors in the simulation.

vi. *Setting up the Variables and Steps.*

The set up includes the rest of the variables in the simulations, and consists of 3 steps. A step is a segment in the simulations, which various conditions are set for a certain execution time. The first phase is the initial step, then the dropping phase followed by a short settling phase. However before setting up the steps a few other interactions have to be set. The first property is the contact property, which is set to hard contact. Then the surface-to-surface contact constraint is set using the surfaces that were created previously when making the parts. In surface-to-surface contact there are two types of surfaces, a slave surface and a master surface. The master surface generally should be the larger surface; because of this the ground surface is assigned as the master surface. Under surface-to-surface contact there are two separate interactions available, kinematic method and the penalty contact method. The method being used is the penalty contact method. This is used since this method takes into account a basic dampening force that allows the part to settle more realistically and creates a friction force on the surface.

Step-1

Step-1 also called the Initial Step is where the initial conditions of the part are inputted and has a runtime of 0 seconds. The first part is allowing for velocity of the dropping part. Once the velocity for the dropping part is allowed, the initial velocity, which can be calculated using the feed angle, drop height and initial velocity, is inputted into Step-1.

Step-2

Step-2 is where the majority of the simulation occurs, this step is 1 second long and during that time the part is dropped and settles on its resting orientation. In order to drop the part successfully certain variable have to be set. The first task is to fix the grounds surface, so it does not move during the simulation, similar to how an actual ground would act. This is done by fixing the bottom edges of the ground with a variable called ENCASTRE. This fixes the nodes with respect to all translational and rotational movement. Gravity is then added to the whole model with a value of -9.8m/s^2. The interactions that were created before during Step-1 are also propagated into this step, with the expectation of velocity since that has already been applied to the part.

Step-3

Step-3 is to ensure a good reading for the contact forces is available. Their importance will be discussed in Section 3.3.1. The step has a runtime of .1 sec and all the previously created variables from Steps 1 and 2 are automatically propagated into the step by ABAQUS. The only difference is that the gravity is set to -2000m/s^2. This is done so the part fully meshes against the ground and creates the contact values that are need to analysis the landing orientation.

vii. *Running the Simulation*

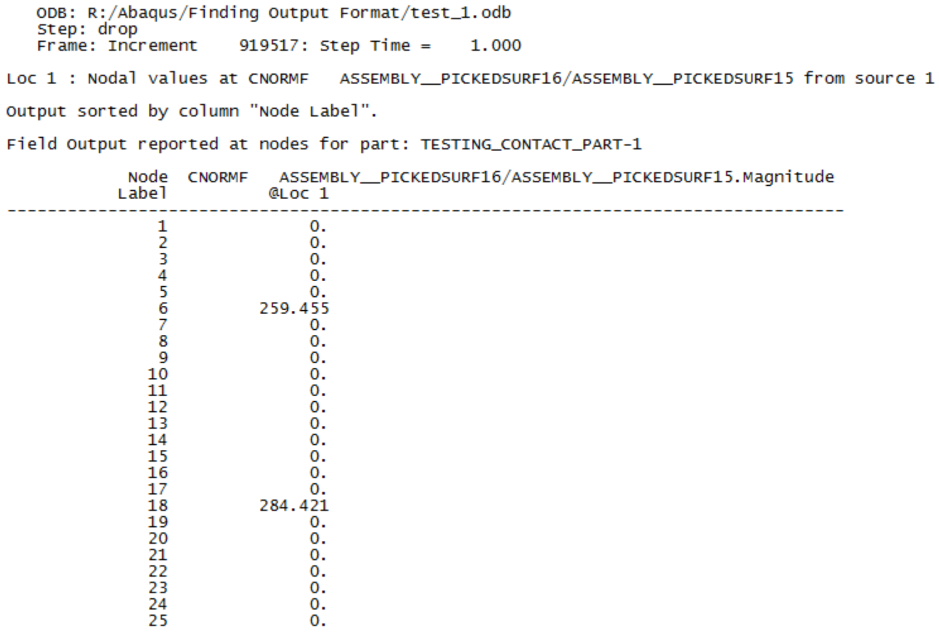
Once the set-up for the simulation is complete a job is then created and submitted. The variables for the job consist of the number of cores and domains. The number of cores is dependent on the size of the model, the quality of the CPU and the CPU's core count. For this project's simulations, four cores were used for a complex model and two for the simple models. The domain has to be a factor of the cores used so if four cores were used in the simulation then the domain value could be any multiple of four. Since this project's simulations are simple with respects to what is possible, the domains were set to the lowest multiple of four.

viii. *Results*

After the completion of the simulation, the results can be obtained. The results can be obtained ether visually in Abaqus' GUI or from cross-referencing input and output files. In order to obtain these results from the output files the following steps must be done.

Gathering Contact Nodes

The first step in obtaining the results are to gather the contact nodes, the contact nodes are the nodes that are currently undergoing any contact forces. In this case, the only possible contact forces would result from the part being in contact with the ground. This is where having the high gravity in Step-3 comes into play. The higher gravity allows for better readings in the contact nodes along with sometimes producing more nodes. The contact nodes can be requested from a field output in Abaqus' GUI. The output format of ABAQUS' contact nodes is shown in Fig. 34.



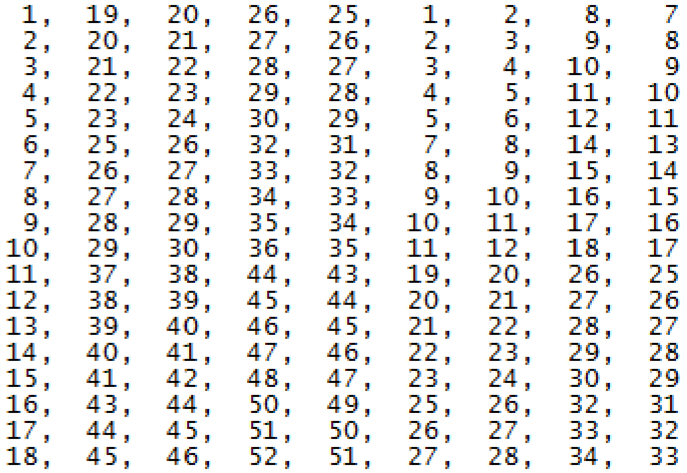
```
   ODB: R:/Abaqus/Finding Output Format/test_1.odb
   Step: drop
   Frame: Increment    919517: Step Time =    1.000

Loc 1 : Nodal values at CNORMF   ASSEMBLY__PICKEDSURF16/ASSEMBLY__PICKEDSURF15 from source 1

Output sorted by column "Node Label".

Field Output reported at nodes for part: TESTING_CONTACT_PART-1

           Node  CNORMF   ASSEMBLY__PICKEDSURF16/ASSEMBLY__PICKEDSURF15.Magnitude
          Label         @Loc 1
----------------------------------------------------------------------------
              1            0.
              2            0.
              3            0.
              4            0.
              5            0.
              6       259.455
              7            0.
              8            0.
              9            0.
             10            0.
             11            0.
             12            0.
             13            0.
             14            0.
             15            0.
             16            0.
             17            0.
             18       284.421
             19            0.
             20            0.
             21            0.
             22            0.
             23            0.
             24            0.
             25            0.
```

Fig. 34. ABAQUS's field output of contacting nodes

Once the .txt file is created by ABAQUS, it is possible to gather every node that is in contact with the ground during the final frame of the simulation. In the case of Fig. 34, the resulting contact nodes were 6, 18, 36, 54, 180, 192, and 198.

Converting nodes to elements

Once the contact nodes have been gathered they have to be converted into the elements the nodes make up. This is done by reading the Input file created by ABAQUS when the job is run. A shortened list of the elements is shown in Fig. 35.



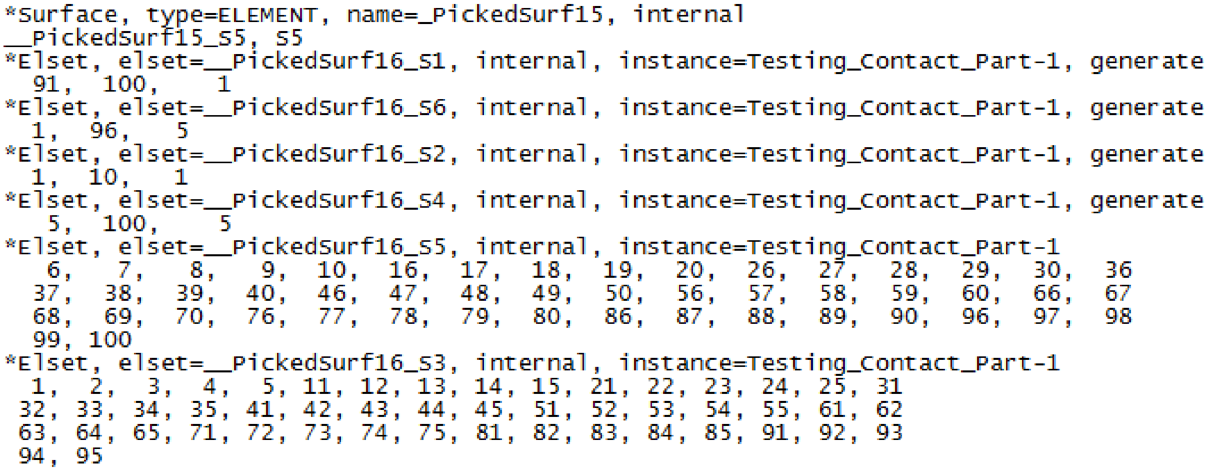
```
 1,  19,  20,  26,  25,   1,   2,   8,   7
 2,  20,  21,  27,  26,   2,   3,   9,   8
 3,  21,  22,  28,  27,   3,   4,  10,   9
 4,  22,  23,  29,  28,   4,   5,  11,  10
 5,  23,  24,  30,  29,   5,   6,  12,  11
 6,  25,  26,  32,  31,   7,   8,  14,  13
 7,  26,  27,  33,  32,   8,   9,  15,  14
 8,  27,  28,  34,  33,   9,  10,  16,  15
 9,  28,  29,  35,  34,  10,  11,  17,  16
10,  29,  30,  36,  35,  11,  12,  18,  17
11,  37,  38,  44,  43,  19,  20,  26,  25
12,  38,  39,  45,  44,  20,  21,  27,  26
13,  39,  40,  46,  45,  21,  22,  28,  27
14,  40,  41,  47,  46,  22,  23,  29,  28
15,  41,  42,  48,  47,  23,  24,  30,  29
16,  43,  44,  50,  49,  25,  26,  32,  31
17,  44,  45,  51,  50,  26,  27,  33,  32
18,  45,  46,  52,  51,  27,  28,  34,  33
```

Fig. 35. Element list obtained from ABAQUS's input file.

The first column in Fig. 35 is the element number, while the remaining columns are the nodes that make up the element. The contacting nodes can be referenced against the list of elements in order to get the contacting elements. In this case the contacting elements are 5, 10, 20, 30, 90, 95, and 100.

Finding the Correct Surface

With the contacting elements known, the next step is to go back into the Input file and reference them against the surfaces that were created. An example of the Input files surfaces is shown in Fig. 36.



```
*Surface, type=ELEMENT, name=_PickedSurf15, internal
__PickedSurf15_S5, S5
*Elset, elset=__PickedSurf16_S1, internal, instance=Testing_Contact_Part-1, generate
  91,  100,    1
*Elset, elset=__PickedSurf16_S6, internal, instance=Testing_Contact_Part-1, generate
  1,  96,   5
*Elset, elset=__PickedSurf16_S2, internal, instance=Testing_Contact_Part-1, generate
  1,  10,   1
*Elset, elset=__PickedSurf16_S4, internal, instance=Testing_Contact_Part-1, generate
   5,  100,    5
*Elset, elset=__PickedSurf16_S5, internal, instance=Testing_Contact_Part-1
   6,   7,   8,   9,  10,  16,  17,  18,  19,  20,  26,  27,  28,  29,  30,  36
  37,  38,  39,  40,  46,  47,  48,  49,  50,  56,  57,  58,  59,  60,  66,  67
  68,  69,  70,  76,  77,  78,  79,  80,  86,  87,  88,  89,  90,  96,  97,  98
  99, 100
*Elset, elset=__PickedSurf16_S3, internal, instance=Testing_Contact_Part-1
  1,  2,  3,  4,  5, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 31
 32, 33, 34, 35, 41, 42, 43, 44, 45, 51, 52, 53, 54, 55, 61, 62
 63, 64, 65, 71, 72, 73, 74, 75, 81, 82, 83, 84, 85, 91, 92, 93
 94, 95
```

Fig. 36: Surface list generated from ABAQUS's Input file

In order to read this, an understanding of how ABAQUS names its surfaces and element lists are needed. As an example the second * in Fig. 36 says Elset, elset=_PickedSurff16_S1……, this can be simplified to S1, which means surface 1. Following that it gives a list of 3 numbers 91, 100, 1. This list means that the surface comprises of elements 91 through 100 by increments of 1. However if the list is longer than 3 it simply means it comprises of those elements. For the example the elements 5, 10, 20, 30, 90, 95, and 100 fall into the surface set of 5, 100, 5, which in results means the part landed on surface 4.

ix. ***Convergence Tests***

As stated in Section 3.3.1.c.iv when running the convergence tests for our simulations, the final landing orientation of the part was the main focus. In supplement to the average distance traveled by each node was taken as another comparison.

Procedure of Convergence Tests

In order to first start the convergence test a base part and ground is needed, for all of our tests this was done by transferring models from SolidWorks to ABAQUS via STEP file. One important thing to make sure is that before the model is exported as a STEP file the model is modeled in meters so it will be consistent with the unit system chosen for ABAQUS. Once the part is saved out as a STEP file it is then easily imported into ABAQUS. Once imported into Abaqus the setup of the simulation can begin. The simulation can be prepared the same way as shown in Section 3.3.1.c, the only difference is the meshing of the part. In ABAQUS's auto meshing feature the mesh size and node count is dependent on the seed size, which relates to the distance between nodes. As the seed size reduces the nodal count will be increasing and therefore decreasing the mesh size. In order to change the mesh size and node count over a range of tests the seed size has to be changed. ABAQUS calculates a suggested seed size that is dependent on the geometry of the model that is imported. This allows for a nice base size in which we can extrapolate in both directions.

Example of Convergence Test with Simple Part

Before the tests and simulations were started on the part shown in Fig. 37, a convergence test on the part had to be ran. This part is the same part that was experimented with using the Theoretical Method in Section 3.2.5.d.

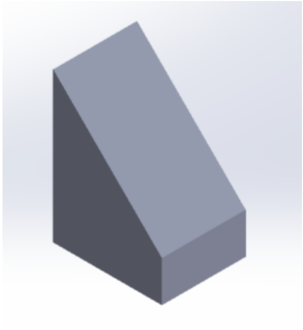


Fig. 37. Initial convergence test part, modeled in SolidWorks

The initial variables that were decided for the base run were an initial velocity of 0.2 m/s, a shoot angle of 90 degrees, and an initial height of 0.2m. The results of these convergence tests can be seen in Table 1.

Table 1. Results of primary convergence test of simple object

Seed Modifier	Node Count	Total Nodal Displacement	Average Displacement	Surface Convergence
x1.3	150	5.72809	0.038187267	no
x1.15	252	8.17573	0.032443373	no
x1	336	14.6159	0.043499702	no
x0.9	502	24.1483	0.048104183	yes
x0.85	616	22.1129	0.035897565	yes
x0.75	864	29.8747	0.034577199	yes
x0.65	1260	41.6643	0.033066905	yes

The seed modifier in Table 25 refers to the value in which the default seed size was multiplied by in ABAQUS. The total nodal distance is the sum of the each individual node’s travel distance from its initial orientation. This value is obtained by using ABAQUS’s field output command for the last frame of the simulation. The output file is then created by the ABAQUS in the form of a .txt file, which can be read using any text software such as Notepad. The average displacement is then calculated from the total nodal displacement and the node count. The final column, which is surface convergence, is whether or not the surface of that specific run matches the surface of the final run, or the run with the highest node count. The results were then graphed in Fig. 38.

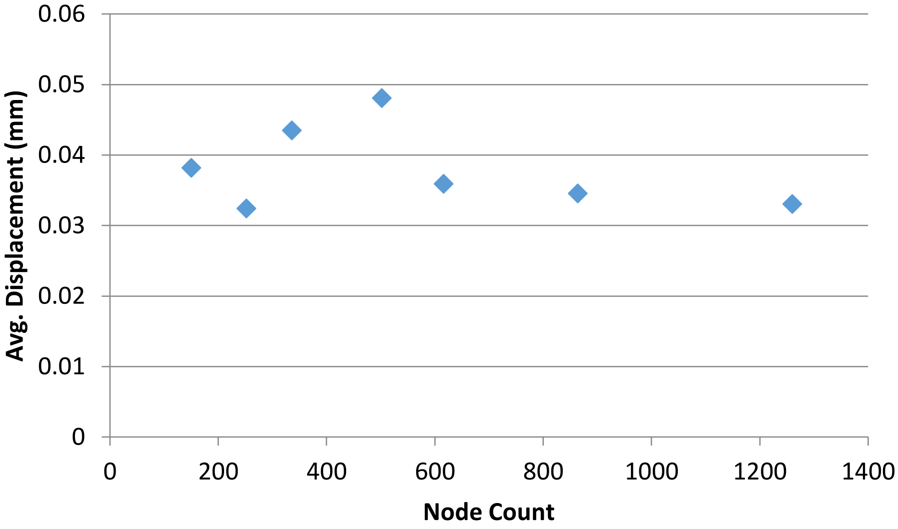


Fig. 38. Result from the convergence test of the simple object.

Upon observation of these results it can be seen that the average nodal distance does begin to converge as expected. However since the only real data that is being obtained from this simulation is the final resting orientation of the part, and that starts to converge around a nodal count of 500. It can be concluded that the part is sufficiently meshed by 600 to 800 nodes.

Given this was the first convergence test and we were unaware how other factors impacted the final convergence of the part other tests needed to be conducted regarding the changing of the initial variables. The variables that are being investigated are velocity, material properties, angle of impact, and initial orientation. In order to do this each variable was both increased and decreased one at a time. So the impact of how it changes the convergence rate can be observed. The results are shown both in Appendix M and in Fig. 39.

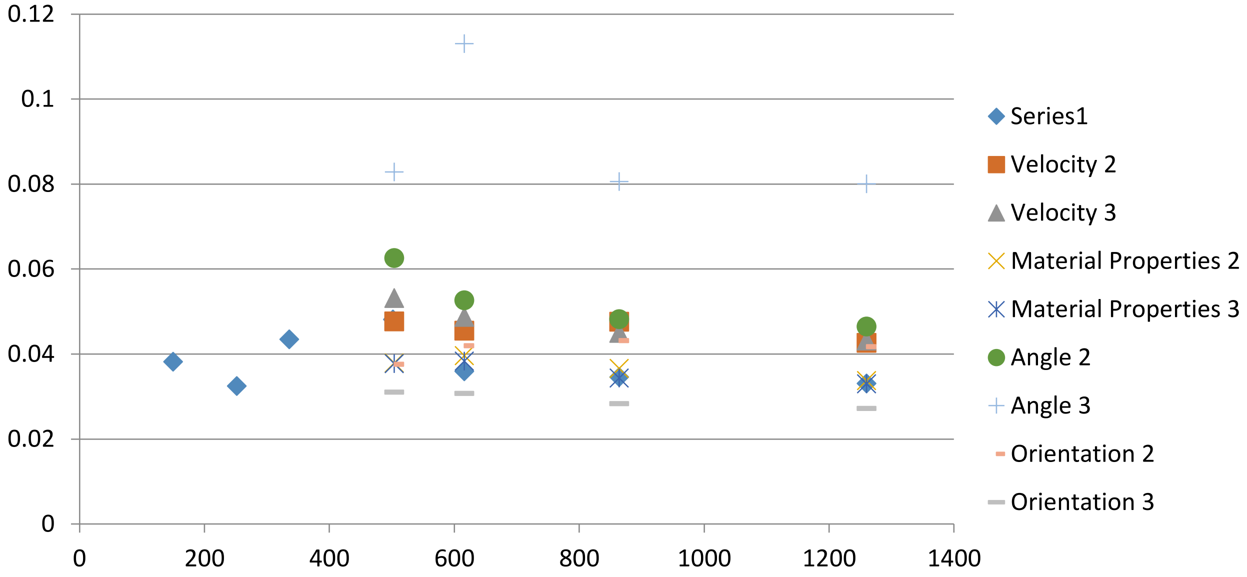


Fig. 39. Graph of all the convergence variables we tested

Through testing it can be concluded that the convergence of the mesh is fairly independent to all the variables that are going to change throughout the tests, this is shown in Fig. 39. However what was observed was that the mesh size was dependent on the material properties of the part. These were the only runs that the final resting orientation of the part did not converge by 616 nodes although the displacement appeared to converge.

3.4. Automation of Computational Simulations

The task of carrying out the procedures outlined and setting the simulations up individually, would not be practical for the objective of this project. Thus, a script was designed and created to fully automate the process after the first input file is created. This method is further outlined in Section 4.2

3.4.1. Python Script

Python, often compared to other interpreted languages such as Java, JavaScript, Perl and so on, is a powerful object-orientated scripting language that is used widely by organizations throughout the world. The use of Python requires minimal programming background from the user. It is also readable, flexible and very expressive.

ABAQUS makes extensive use of Python, which has been embedded within the ABAQUS software products. The language extensions are referred to as the ABAQUS Scripting Interface (ASI), which is an application-programming interface (API) to the models and data used by ABAQUS, and the finite element analyst at many different levels can use it. Generally, ABAQUS Scripting Interface scripts are Python scripts.

At a basic level, the scripts can be used to automate repetitive tasks such as the creation and submission of ABAQUS analysis jobs. From a single run analysis standpoint, the script can eliminate the procedures that would require users to interact with the ABAQUS GUI. From multiple run analysis standpoint, it can eliminate all the repetitive procedures thus reducing the overall time input required from the users. Increased efficiency and minimization of human errors are two major advantages of utilizing a script.

3.4.2. ABAQUS PDE

ABAQUS Python Development environment (PDE) is an application in which users can create, edit, test, and debug Python scripts. It is a separate application that can be accessed from within ABAQUS /CAE or launched independently to work on Python scripts. It is primarily used with scripts that use the ABAQUS /CAE graphical user interface (GUI) or kernel commands, including plug-ins, but it can also be used to work on scripts that are unrelated to ABAQUS. The ABAQUS PDE also enables one to set breakpoints to pause script execution at a particular line in any Python script. In this project, the python script is developed under ABAQUS PDE.

3.4.3. Development of Automation Script: Overview

The whole automated process will be completed by two types of scripts: Python script and MATLAB script.

The Python script was used to automate the FEA procedures. It was used to completely eliminate any direct interaction with ABAQUS from the user, only requiring some prompt inputs. The Python script was used to create a series of ABAQUS input and output files.

The MATLAB script was mainly used to extract the series of data obtained from the input and output files from ABAQUS and cross-reference each other to get the final results. The final output is a histogram showing the probability distribution final landing surfaces. The desired overall flow of automated process is shown in Fig. 40.

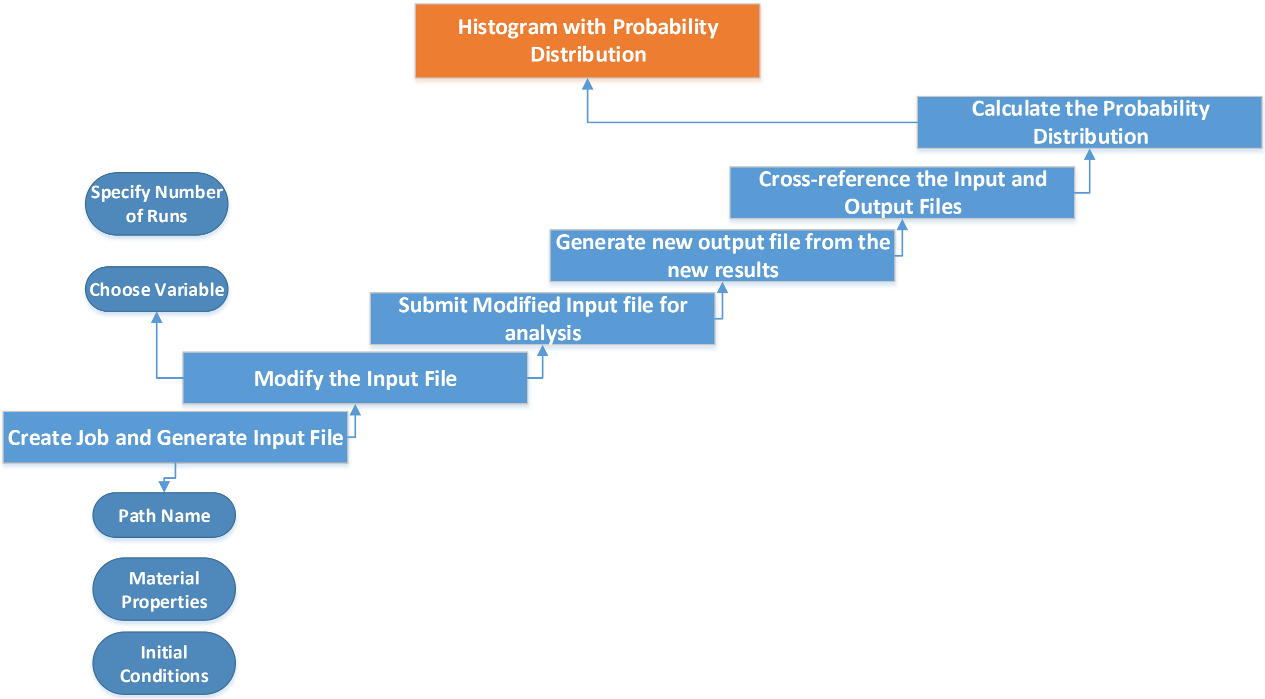

Histogram with Probability Distribution
Calculate the Probability Distribution
Cross-reference the Input and Output Files
Specify Number of Runs
Generate new output file from the new results
Choose Variable
Submit Modified Input file for analysis
Modify the Input File
Create Job and Generate Input File
Path Name
Material Properties
Initial Conditions

Fig. 40. Overall Scrip Flow

3.5. Experimental

3.5.1. Physical Testing Procedure

Physical testing was conducted on all the objects used to analyze the validity of the theoretical and computational tools. For each object, the calculation for the sample size was conducted to find the minimum number of drops for each object so that the results can be said with 90% confidence that it is accurate. Details on the calculation of sample size can be found in

Appendix D. Sample Size Calculation and the sample size for each object can be found in Table 16 of the same section.

Physical testing was conducted by randomizing the orientation and dropping the part at a 90 degree angle, 400mm from the flat ground then observing the landing orientation and recording the results. The part is dropped one at a time to ensure that there were no interactions between the parts during the drop, which may affect the outcome of the result. The testing was conducted indoors without environmental factors such as wind, which could affect angle of projectile.

3.5.2. High-speed Imaging

In order to further examine how parts orient themselves and the physics behind dropping, high-speed footage and data was collected. The footage was acquired using a high speed camera from Correlated Solutions at around 6,000fps with a resolution of around 700x1000. This was done mainly to assess the impact of how having multiple material properties could impact the probabilities of the parts final resting orientation. For these tests an object with multiple material properties was dropped from a height of 20 cm. The first data set was taken when the part landed on the rigid side, while the second test was taken for when the part landed on its elastic side. During the drop using VIC-3D, software provided by Digital Image Correlation, the two data points were able to be tracked during the fall. These data points gave us the ability to track the parts position in three dimensional spaces as the part fell using control points on opposite corners of the part.

Using the set of data points obtained from the footage, 3D graphs of the parts control points were graphed using Techplot 360. This allows us to see how the part falls through space. These graphs are shown in Fig. 41.

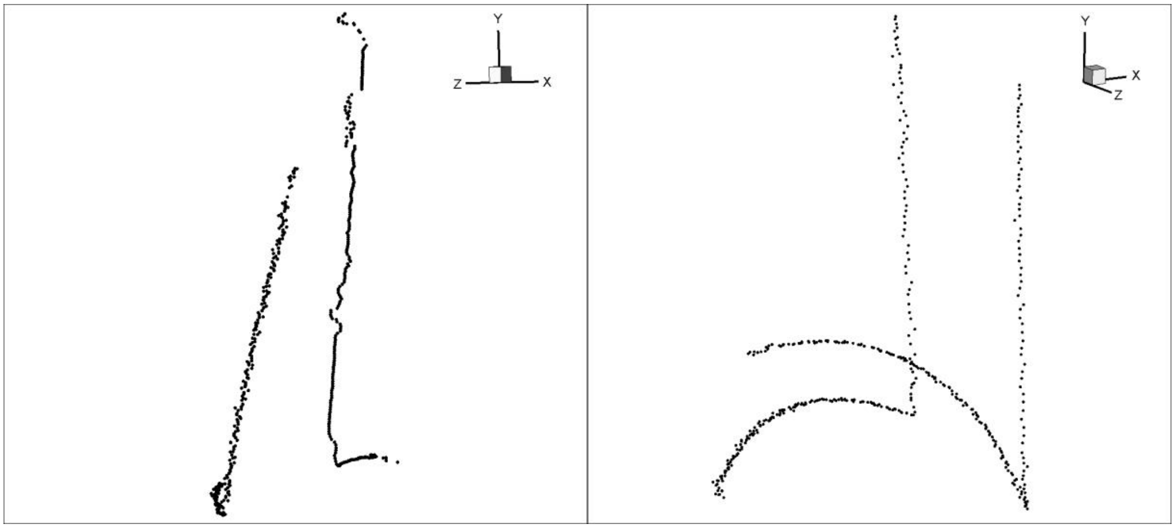


Fig. 41. 3D position plots of control points. Graph (a) on left shows the first test of the ridged landing surface; Graph (b) on right shows the results of the part when landing on its elastic surface.

During tests, due to the limited resolution of the camera it was not possible to get the full drop of the elastic surface in frame. This can be seen in Fig. 41 (b), the reason the graph does not end with the part on the ground is because it bounces out of frame. From the position data that was obtained from the cameras software, and the time that can be obtained via frame rate and frame number, the velocities were calculated. These results are shown in Fig. 42, 43, 44 and 45.

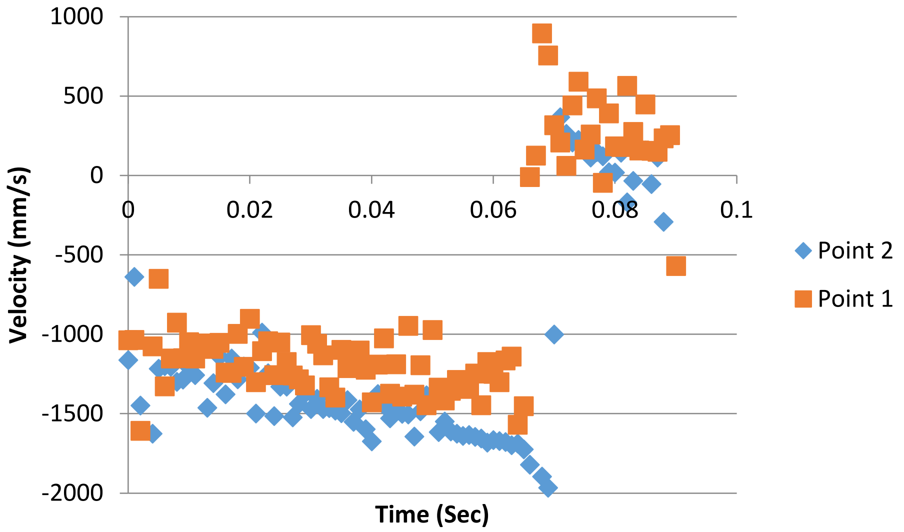


Fig 42. Graphical representation of velocities in the y direction of control points 1 and 2 during test 1, where the part lands on its rigid surface.

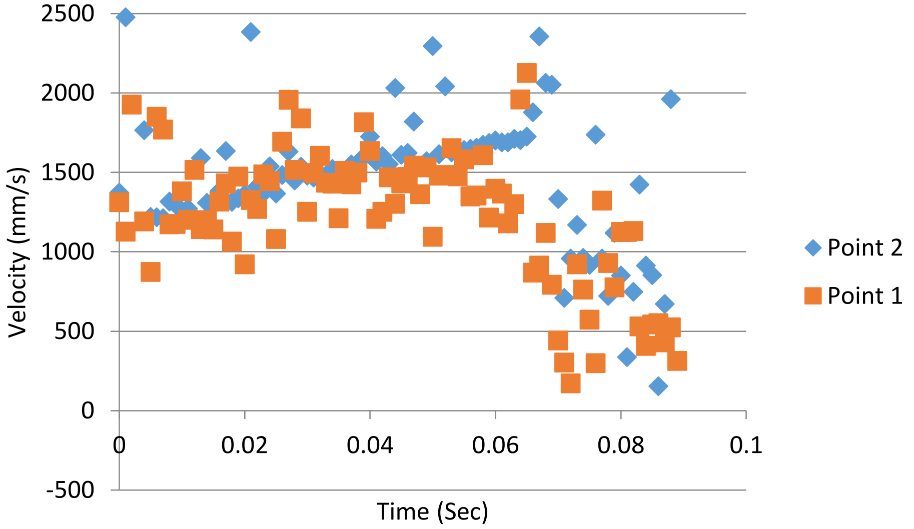


Fig 43. Graphical representation magnitude of control points 1 and 2's velocity during test 1, where the part lands on its rigid surface.

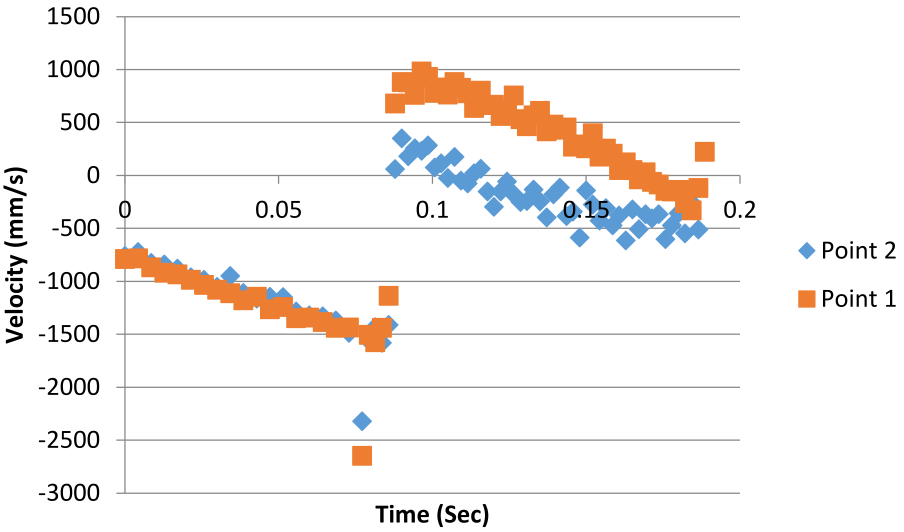


Fig 44. Graphical representation of velocities in the y direction of control points 1 and 2 during test 2, where the part lands on its elastic surface.

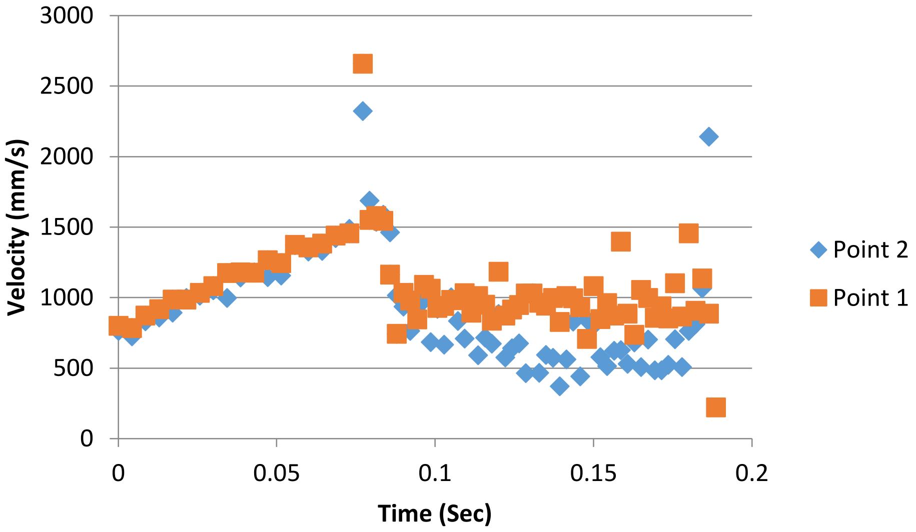


Fig 45. Graphical representation magnitude of control points 1 and 2's velocity during test 2, where the part lands on its elastic surface.

From observations that can be gathered from Fig 43, it is shown that during test one the part took approximately 0.01 seconds to settle. While from observing Fig 45 it can be concluded that the part takes over 0.1 sec to settle. While the actual resting time of the part is unknown for the second test, it can still be concluded that it takes at least a factor of 10 times longer than test 1. In addition to this, the part’s y velocity jumps as high as 1m/s upwards after the bounce as opposed to only around 0.4m/s for when the part lands on the rigid surface.

Although the data did not give any conclusive data able to conclude about how this affects the parts landing surface, what can be concluded is that when a part lands on a more elastic surface it takes longer to stabilize. Along with that it can be hypothesized that if a part has multiple material properties, and faces with the same size and distance from the center of gravity but different properties on those faces, specifically Young’s Modulus. It is likely that the face with the lower young’s modulus is less likely to be its final orientation. This is based on the fact that when a part lands on that face it is provided with enough energy to bounce and “re-shuffle” its orientation. When a part lands on a more rigid surface it is not provided with that opportunity and falls to rest faster.

4. Deliverables

4.1. Theoretical Tool

The theoretical proportion of the project is reliant upon Approach 1 (which also takes into account the Modified Stability Method in Section 3.2.4) of the New Modified Stability Method. The original equation, Equation 1 is used to express the mathematical relationship of the theory. However, after applying the original theory with more complex parts, it is evident that the variables considered in the equation need to be developed. Thus, the algorithm and its implementation are based on the Approach 1 of the New Modified Stability Method.

4.1.1 Algorithm

The flow chart in Fig. 46 shows the algorithm for the theoretical portion of the project. The flow chart starts at the bottom right of the figure. The green process boxes symbolize the inputs of the variables required to find the percentage occurrence of each landing surface. This portion of the process works with objects with any number of surfaces. However, after the Change Distribution decision that is made by the user, the program will only work for objects that can be simplified into a cuboid, as the script was developed to be used specifically with the product of interest. The simplification process and the equations involved are detailed in Section 3.2.6.

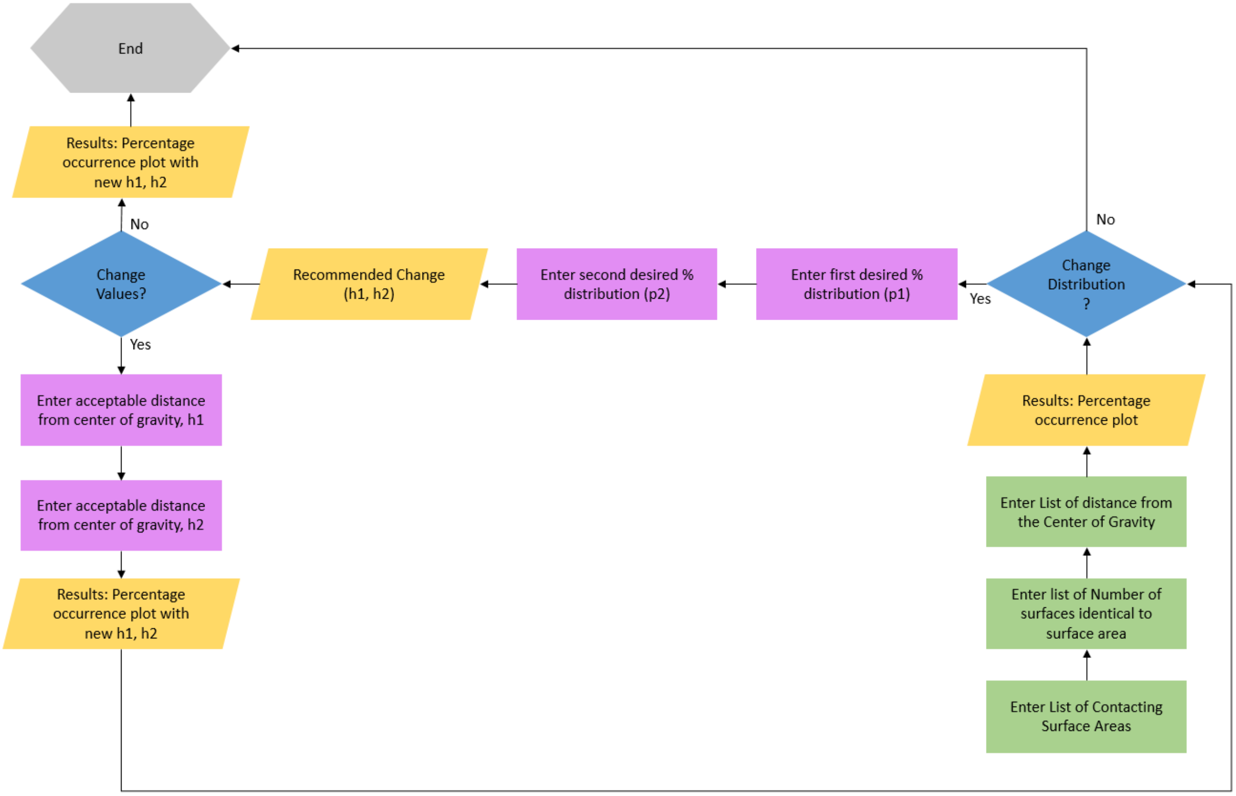

End
Results: Percentage occurrence plot with new h1, h2
No
Change Values?
Yes
Enter acceptable distance from center of gravity, h1
Enter acceptable distance from center of gravity, h2
Results: Percentage occurrence plot with new h1, h2
Recommended Change (h1, h2)
Enter second desired % distribution (p2)
Enter first desired % distribution (p1)
Yes
No
Change Distribution ?
Results: Percentage occurrence plot
Enter List of distance from the Center of Gravity
Enter list of Number of surfaces identical to surface area
Enter List of Contacting Surface Areas

Fig. 46. Algorithm flow chart of theoretical method based on the New Modified Stability Method

4.1.1. Implementation of Algorithm

The algorithm was implemented using MATLAB. A script was developed to mimic the process shown in Fig. 46. The first portion of the script is the function used to calculate the probability of occurrence using Equation 1. This was developed to be flexible so that the first portion of the algorithm can be used with objects that consist of varying number of stable landing surfaces. The script is not included in this paper but a soft copy of the script was provided to the sponsor and the project advisor.

Process enumeration was used and the second portion of the script was developed based on the idea that there are three main stages in the algorithm, each stage involving the user's decision. Stage One is the user's decision to change the distribution, Stage Two is the user's decision on whether the values of h_1 and h_2 are acceptable and Stage Three is the user's decision to end the script. The relationship between the three stages is shown in Fig. 47.

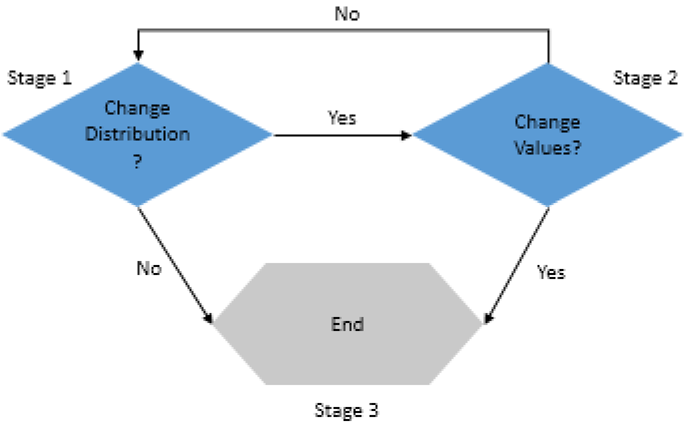


Fig. 47. Stages in the implementation of the theoretical portion

A MATLAB built in function, fsolve was used to solve the simultaneous equation to find the suggest values of h_1 and h_2 based on the desired percentage distribution entered in by the user.

4.1.2. How to use the tool

Two programs are needed to implement the theory based tool, any CAD program capable of find centers of gravity and areas, and MATLAB 8.1 (R2013a); the two tools do not have to be on the same computer. SolidWorks is used to find the surface areas of the landing surfaces and its distance from the center of gravity and MATLAB is used to calculate the percentage occurrence based on the values found using SolidWorks. The user manual, including the general rule on which surface areas should be considered, how to use the measurement tool in SolidWorks and how to use the MATLAB script can be found in Appendix J. User Manual for Theory based tool

4.2. Computational Tool

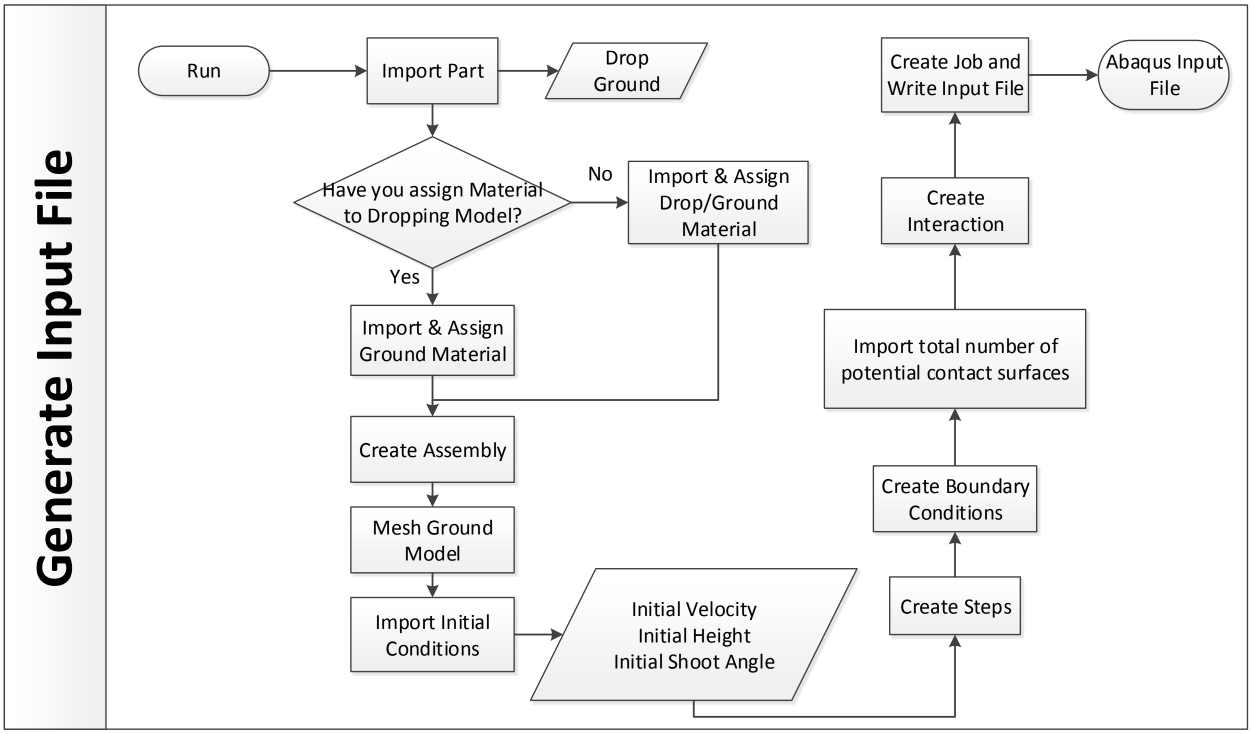
Two programs are required for the computational portion of this project. The first program is a Python script used in ABAQUS Software, which focuses on performing all the FEA procedures automatically and gathering the analysis results. The second program is a MATLAB script that is used to perform data analysis and provide probability distribution of final part landing orientation. Prior to the use of the Python Script, the part needs to be meshed. Instructions on how to mesh the part can be found in Appendix N.

4.2.1. Python Script

a. Algorithm

The overall Python script flow can be divided into three main sections. The first section is to finish a single run of dropping analysis and generates the first ABAQUS input file. The second section is to submit multiple analyses by modifying the original input file. Last section is to generate a series of output files from completed analysis. A complete Python script can be found in Appendix K.

For the first section, the script is commands based and the main flow is shown in Fig. 48. The script flow is the same as the user manually finishing all FEA procedures for a single run of dropping analysis. A detailed explanation and setup can be referred to in Section 3.3.1.c



Generate Input File
Run
Import Part
Drop Ground
Have you assign Material to Dropping Model?
No
Import & Assign Drop/Ground Material
Yes
Import & Assign Ground Material
Create Assembly
Mesh Ground Model
Import Initial Conditions
Initial Velocity
Initial Height
Initial Shoot Angle
Create Steps
Create Boundary Conditions
Import total number of potential contact surfaces
Create Interaction
Create Job and Write Input File
Abaqus Input File

Fig. 48. Python Script First Section Flow

For the second section, the script is more customized to the user's requirement. Several options will be provided for user to choose from. The flow for the second section is shown in Fig. 49. Since the dropping was a randomized process, the part initial dropping orientation should be randomized as well. This is critical because the results obtained from the FEA analysis result will identical if the part's initial dropping orientation is not varied. Therefore the part initial dropping orientation must be randomized to obtain a valid probability distribution. Typically, the script will provide the user a good prediction of the part final landing surface by randomizing the initial dropping orientation. However, the script is also designed to vary the three variables (Initial velocity, height and shoot angle), which controls the part final impact velocity. This allows the user to investigate other dropping initial conditions. For whatever option the users choose, the specific content of original input file will be changed to the value required by the user input.

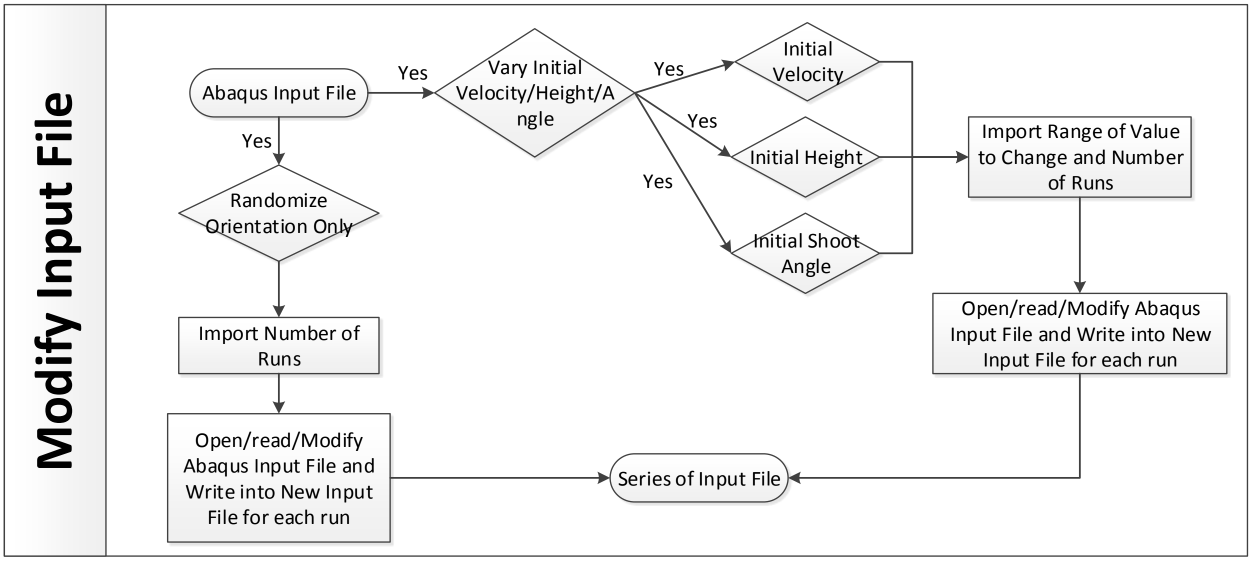

Modify Input File
Abaqus Input File
Yes
Vary Initial Velocity/Height/A ngle
Yes
Initial Velocity
Yes
Initial Height
Yes
Initial Shoot Angle
Import Range of Value to Change and Number of Runs
Yes
Randomize Orientation Only
Import Number of Runs
Open/read/Modify Abaqus Input File and Write into New Input File for each run
Open/read/Modify Abaqus Input File and Write into New Input File for each run
Series of Input File

Fig. 49. Python Script Second Section Flow

In order to change the input file, an understanding of the input file format is needed. The ABAQUS input file composes of a number of option blocks. Each option block begins with a keyword line (starts with *), which is usually forward by one or more data lines. So several functions are written to open and read in specified input file, and then modify the content by finding the keyword lines. The modified input file will be saved as a new input file. The naming of input file will be ordered in number. For example, the number of runs specified by the user will correspond to the number of input files generated.

After a series of input files is generated, the Python script will submit all the input files for analysis. The last section of the Python script is to request field output files from the analyses results as shown in Fig. 50.

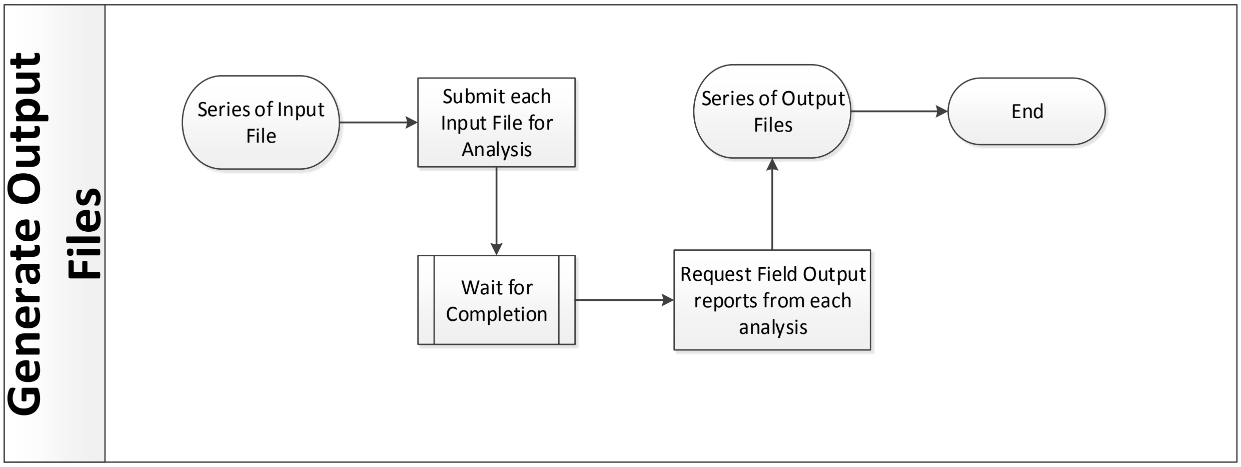


Fig. 50. Python Script Third Section Flow

b. Implementation of Algorithm

Since the Python script was designed to completely eliminate any direct interaction between the user and ABAQUS GUI, several user input is required for a successful analysis.

The object can be any user specified part model. The user needs to provide the path name to the model as an input. Another important input is the material properties for the object. Because the material properties can be important to the final interaction after impact, the user is able to input the specific material property value (Young’s modulus, passion ratio and density) of the object.

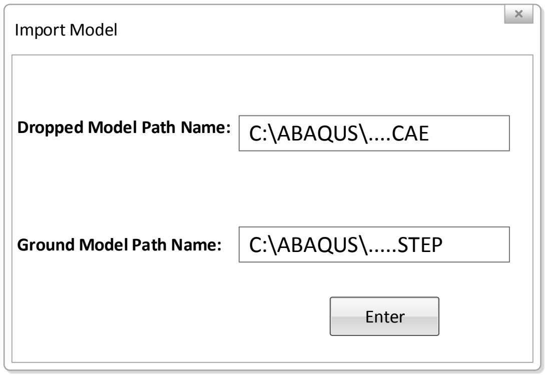
The initial velocity of drop is another important parameter. In order to calculate the object’s final impact velocity, several parameters needs to be taken into account. These parameters are: initial velocity, initial height and initial shoot angle. A function is used to calculate the final impact velocity based on these parameters, and the calculated impact velocity is being used in the analysis. Therefore, the user needs to provide the initial conditions of those three parameters. These values should depend on the setup of realistic drop testing. The user also needs to input the number of labeled surfaces in the part to allow the script select all the surfaces for further contact analysis.

Two ways of generating input files will be provided to the user: randomize orientation only or vary initial velocity/height/shoot angle. Since the repeated analysis is achieved by submitting a series of modified input files, the user needs to specify the number of runs required. This number will be taken by the Python script to generate specified number of input files. If vary initial velocity/height/shoot angle is chosen, a range of value needs to be specified by the user. The script will ask for a start value and an end value, in addition with the number of runs. Therefore, the series of input files generated will cover the whole range of value that used specified.

c. How to use the tool

Before running the script, the user needs to make sure the followings are completed: (a): the part has been accurately meshed (instructions can be found in Section 3.3.1.c.ix); (b): the contact surfaces have been defined and labeled (instructions can be found in Section 3.3.1.c.iv); c) Assign the material properties to the dropping model. Assigning the material properties is an optional requirement. If the user chooses to not assign the material properties to the model, there is still an opportunity to assign the material properties it in the following automation procedure.

After all the prerequisites are met, the user can launch the ABAQUS software, set up the working directory and open ABAQUS PDE. Within ABAQUS PDE, open and run the Python script. Several inputs are required from user while running the Python script. Step 1 is to import models, as shown in Fig. 51. The user needs to provide the path name to the dropping object and the ground model.

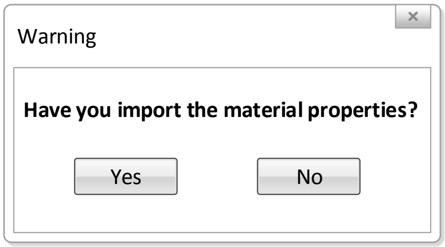

Import Model

Dropped Model Path Name: C:\ABAQUS\....CAE

Ground Model Path Name: C:\ABAQUS\.....STEP

Fig. 51. Step 1 Import Models

Step 2 as shown in Fig. 52, is a warning message to let the user select whether to import material properties. The user can select “Yes” if they had previously assigned the material properties in the imported model. The user can select “No” if they have not previously assigned the material properties in the imported model.

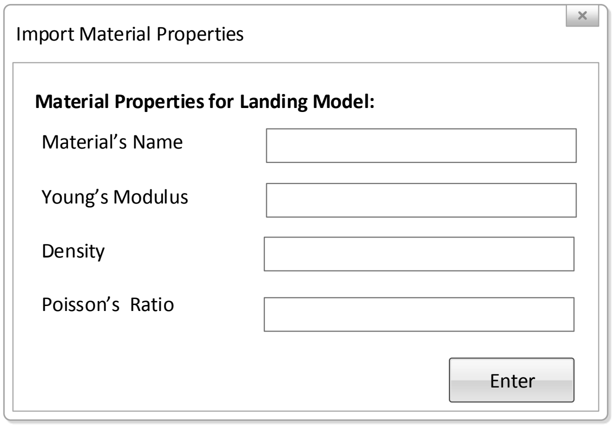


Warning

Have you import the material properties?

Fig. 52. Step 2 Warning Message

Step 3, shown in Fig. 53, is to let the user import the material properties only for ground model if ‘Yes’ was selected in Step 2. If the user selects “No” in Step 2, the user needs to import the material property for both dropping and landing model, which is shown in Fig. 54.



Import Material Properties

Material Properties for Landing Model:

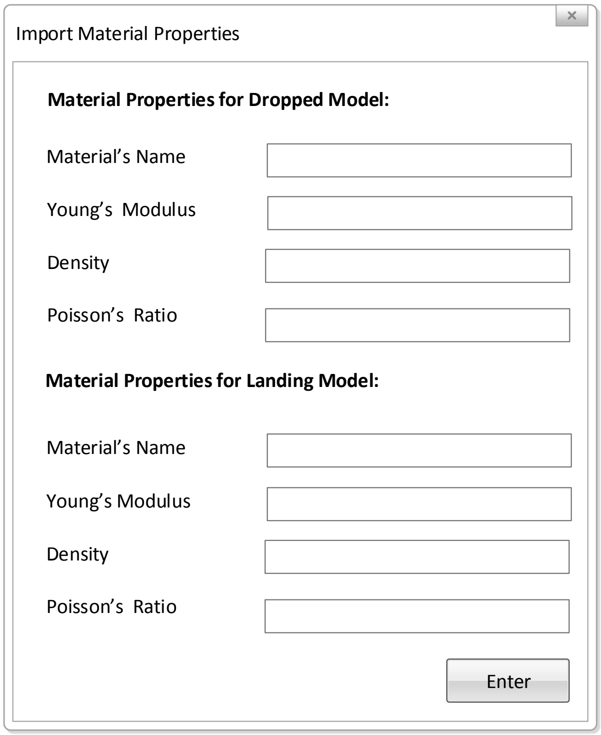
Material’s Name

Young’s Modulus

Density

Poisson’s Ratio

Fig. 53. Step 3 Imports Material Properties if ‘Yes’ Selected



Import Material Properties

Material Properties for Dropped Model:

Material’s Name

Young’s Modulus

Density

Poisson’s Ratio

Material Properties for Landing Model:

Material’s Name

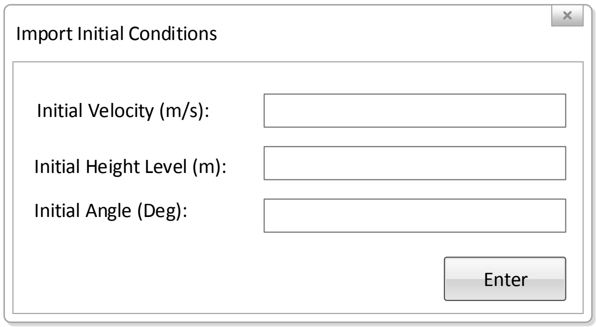
Young’s Modulus

Density

Poisson’s Ratio

Fig. 54. Step 3 Imports Material Properties if ‘No’ Selected

Step 4, shown in Fig. 55, is to import initial conditions. The user is required to input three values in this step. The Initial Velocity will be a negative value since the velocity direction is pointing downward to the ground, e.g. -0.1. Put in 0 if there is no initial velocity involved. The Initial Height will be a positive value, e.g. 0.2 and the Initial Angle will be any value in a range of 0 – 90 (0 and 90 are excluded). If it’s a simply dropping procedure, then put 90 for initial angle.



Import Initial Conditions

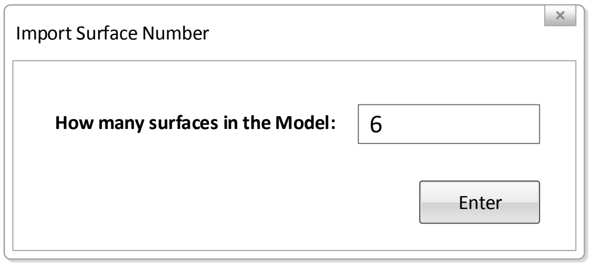
Initial Velocity (m/s):

Initial Height Level (m):

Initial Angle (Deg):

Fig. 55. Step 4 Import Initial Conditions.

Step 5 shown in Fig. 56 allows the user to import the total number of labeled surfaces the part has. Only the surfaces which has possibility of contacting with the ground after landing will be labeled. Therefore the number of labeled surfaces means the number of surfaces that are potentially can be in contact with the ground after landing. An explanation can be found in Section 3.3.1.c.iv and a detail instruction of how to label the surface can be found in Appendix N, step 12-15. The user is required to enter an integer value, e.g. 6. This step is critical for an accurate analysis.



Import Surface Number

How many surfaces in the Model: 6

Fig. 56. Step 5 Import Labeled Surface Number.

After Step 5, the first ABAQUS input file is generated. The following procedures will require inputs from the user to generate a series of new input files by modifying the original input file. Step 6 is to import parameter as shown in Fig.57. If the user wants to see the landing orientation results with a specified initial condition, the user can select “Randomize Initial Orientation Only”. Then the user will be required to enter an integer value, e.g. 20. This number is the number of runs required by the user, which is shown in Fig.58.

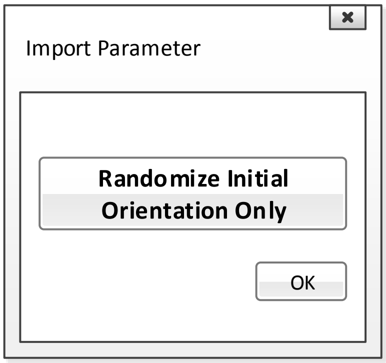


Fig. 57. Step 6 Import Parameter

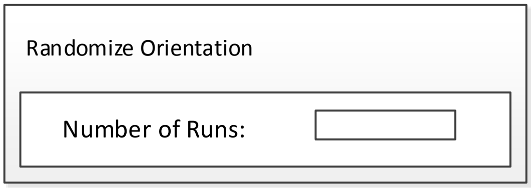


Fig. 58. Randomize Orientation

After step 5, a series of input file will be generated and submitted for analysis. When the analyses are done, a series of output files will be requested and generated automatically. This completes the Python script portion of the computational method. Overall detailed instructions on how to use Python program can be found in Appendix K.

4.2.2. MATLAB Script

a. Algorithm

A MATLAB script is used to extract the data from the ABAQUS input and output files. Because there is a large amount of data to read from series of input and output files, MATLAB can be utilized to make it more efficient to obtain a customized result. The extracted data will be saved into matrices, thus it will save time while referring to specific data. A complete MATLAB script is not provided in this report.

The MATLAB script can be divided in four sections. The first section is to open and read the ABAQUS input file, as shown in Fig.59. From the input file, under dropping object nodal definition section, all the nodes and its corresponding element number will be read and saved into matrix “Node”. All the labeled potential contact surfaces in dropping object will have its corresponding set of element number. The data for each labeled surface will be extracted and saved into matrix “Surface”. And then this matrix “Surface” will be reorganized to let the matrix row number represent the surface number. Therefore all the elements for a surface will be placed into one row. The new matrix is named as “DistributedSurface”.

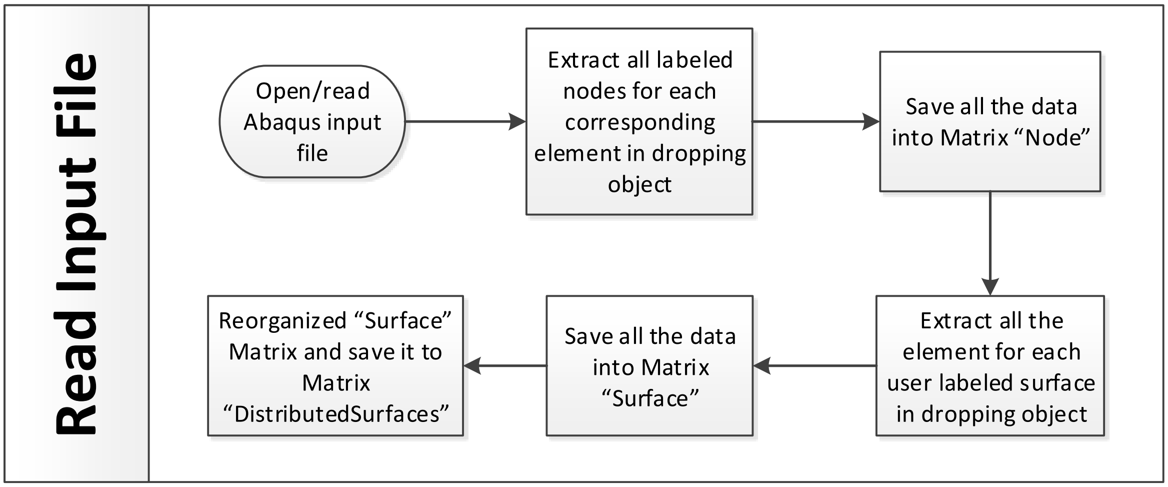

Read Input File
Open/read Abaqus input file
Extract all labeled nodes for each corresponding element in dropping object
Save all the data into Matrix "Node"
Extract all the element for each user labeled surface in dropping object
Save all the data into Matrix "Surface"
Reorganized "Surface" Matrix and save it to Matrix "DistributedSurfaces"

Fig. 59. MATLAB Script First Section Flow

The second section is to open and read in the ABAQUS output file, as shown in Fig. 60. All the node number, where its corresponding contact force is non-zero, will be extracted and saved into matrix "Element".

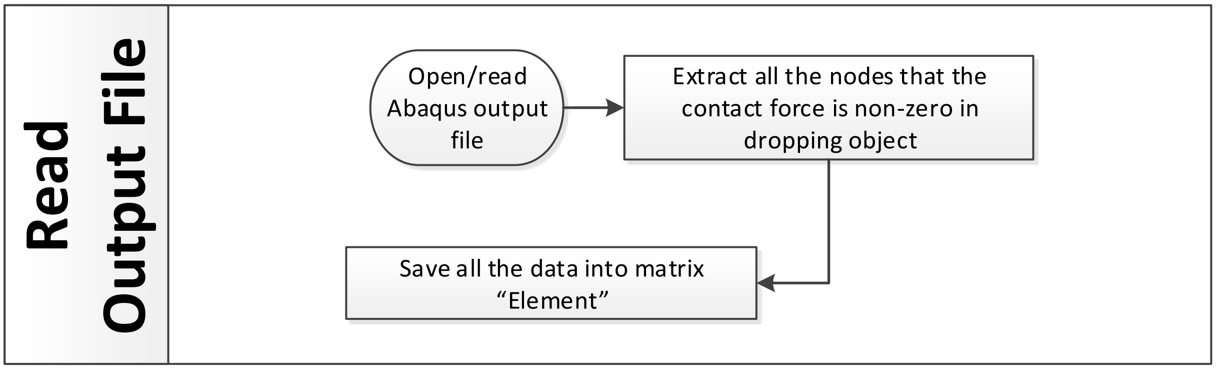
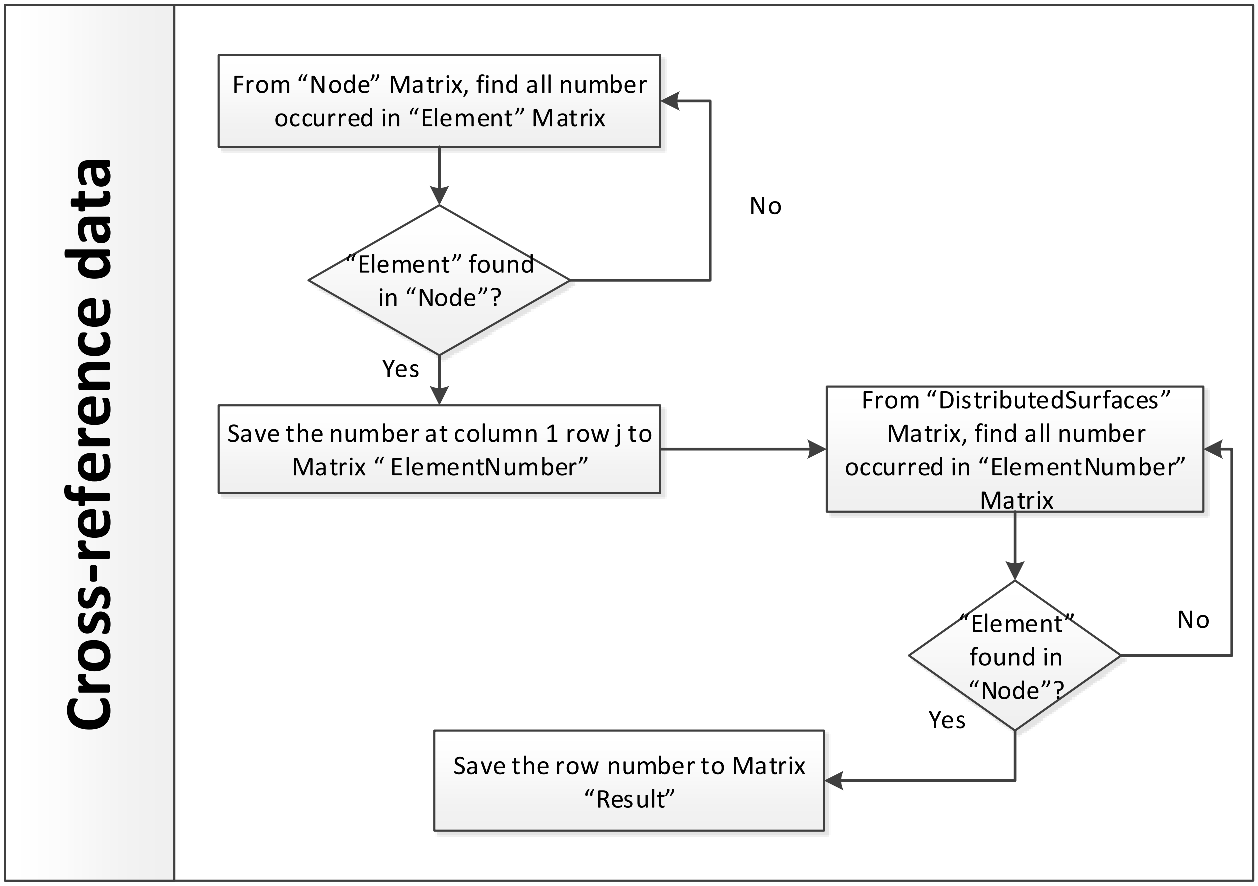

Read Output File
Open/read Abaqus output file
Extract all the nodes that the contact force is non-zero in dropping object
Save all the data into matrix "Element"

Fig. 60. MATLAB Script Second Section Flow

The third section is to cross-reference the data extracted from the input and output files, as shown in Fig. 61. Matrix "Node" and matrix "Element" will be compared first. If the number from

“Node” occurs anywhere in “Element”, the corresponding number in column 1 at that row will be saved into matrix “ElementNumber”. Then matrix “ElementNumber” and matrix “DistributedSurface” will be compared. If all the numbers from “ElementNumber” are found in “DistributedSurface”, the corresponding row number will be saved into matrix “Result”.



Cross-reference data
From “Node” Matrix, find all number occurred in “Element” Matrix
No
“Element” found in “Node”?
Yes
Save the number at column 1 row j to Matrix “ ElementNumber”
From “DistributedSurfaces” Matrix, find all number occurred in “ElementNumber” Matrix
“Element” found in “Node”?
No
Yes
Save the row number to Matrix “Result”

Fig. 61. MATLAB Script Third Section Flow

In the last section, the result obtained from each analysis will be gathered and then calculated for a final landing surface probability distribution. The script flow is shown in Fig. 62. The final results will be displayed using a histogram.

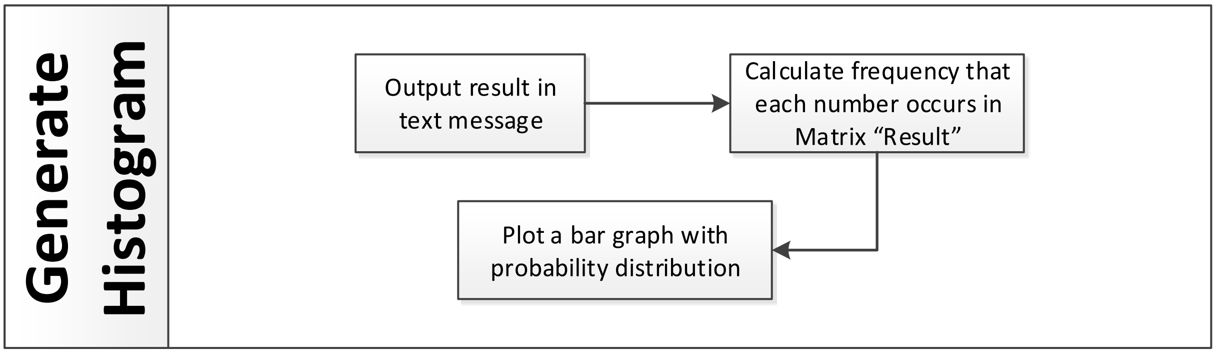

Generate Histogram
Output result in text message
Calculate frequency that each number occurs in Matrix "Result"
Plot a bar graph with probability distribution

Fig. 62. MATLAB Script Last Section Flow

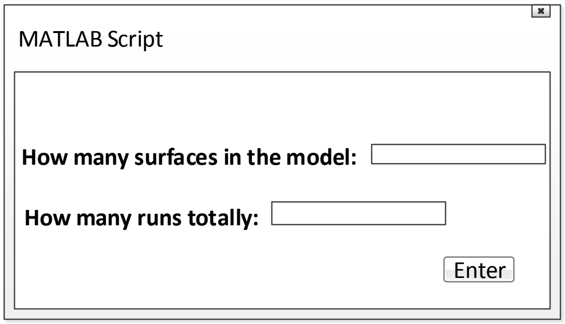
b. *Implementation of Algorithm*

In order to read in all the input and output files, the user need to specify the total number of analysis performed in ABAQUS. This will help the MATLAB script read in all the files needed. Another input required from the user is the total number of labeled surfaces the object has. This is the same number input as explained in Section 4.2.1.c Step 5. This input notifies the MATLAB script the number of data values it needs to extract from the ABAQUS Input files.

c. *How to use the tool*

Before running the MATLAB script, the user needs to ensure that one input file title with "Drop_test.inp" and all the output files (files name ending with .rpt) are available in the same working directory as the MATLAB script.

After running the MATLAB script, two inputs are required from the user as shown in Fig. 63. These inputs need to be consistent with the previous Python script steps. First the user needs to enter the inputs as in step 5 of the running Python script again. Second the user needs to enter the input as in step 6 of running Python script again, which should be the total number of runs, or the number of output files available in the working directory.



MATLAB Script

How many surfaces in the model:

How many runs totally:

Fig. 63. MATLAB User Inputs

The end result of the MATLAB script is a histogram similar to Fig.64, and a text message with detailed results will be shown in the MATLAB command window.

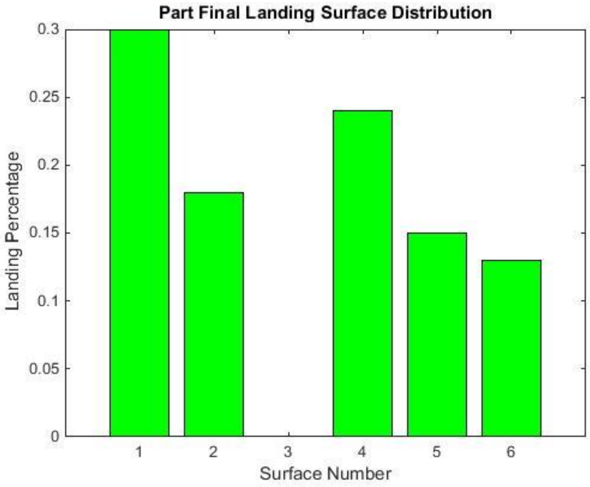


Fig. 64. MATLAB Script Result Sample

4.3. Validation of the tools

The validation of the theoretical tool is described in Section 3.2.5. In this Section, details of the validation of the computational tool are provided, including: (a) comparing results of simple model with physical testing; (b) variable dependency analysis to investigate the effects of shooting angle, drop height, velocity in drop test; and (c) the application to the simplified model of the product of interest In addition, the theoretical and computational tools were validated with physical testing.

a. Test Results for Comparison to Physical

Tests were ran with the simple model (Object 4) shown in Fig. 65, through our simulation with an initial height of 0.2m, shoot angle of 90 degrees and velocity of - 0.1m/s. After the simulations' completion the following results were obtained. Since this run was also used in order to test for appropriate sample size 100 runs were done. Surface 1.1 and Surface 1.2 have been combined in the graph since they are symmetrical and can be considered the same, while Surface 5 is unstable so it will never land on it so it is not displayed on the graph. Therefore, only four final surfaces are considered while interpreting the results. Surface 1.1 and 1.2 will be considered as Surface 1. Surface 5 will not be considered.

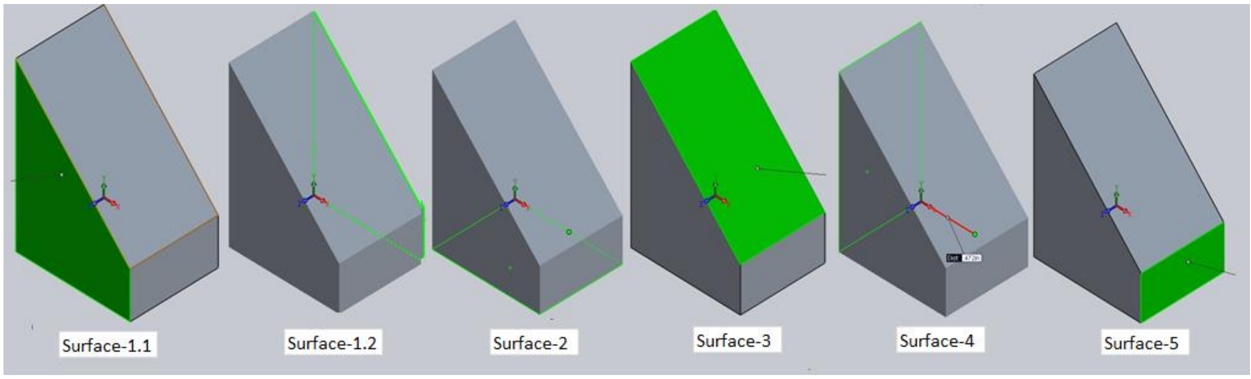


Fig. 65. Simple Object Surface Labels

Thus there will be totally four possible orientations when the simple model rests after dropping. Orientation 1 will be the model landing on either Surface 1.1 or Surface 1.2. Orientation 2 will be the model landing on Surface 2. Orientation 3 will be the model landing on Surface 3 and Orientation 4 will be the model landing on Surface 4. The computational results of dropping simple model in comparison to the physical testing are shown in Fig. 66.

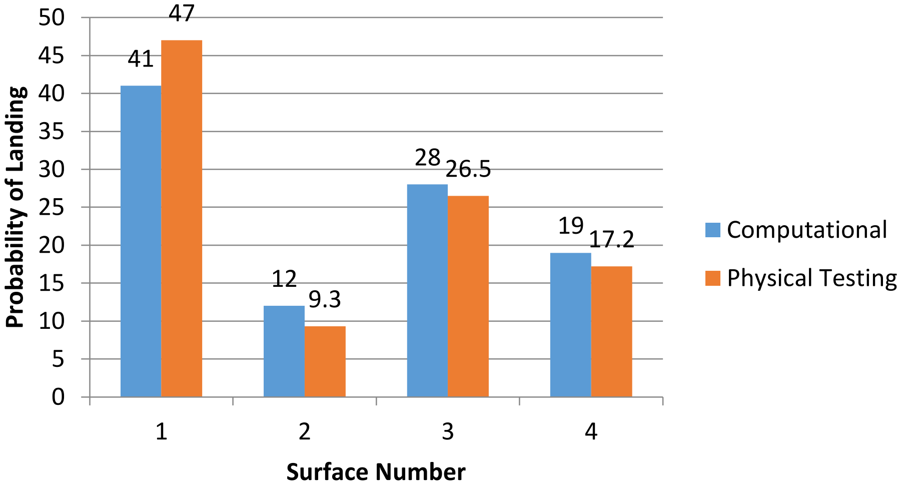


Fig. 66. Graphical representation of computational test results vs. physical test results.

After comparing the computational data to the physical results, an average percent error of 14.5% was computed

b. Variables Dependency Analysis

The simple model mentioned in Section 4.3.a is used to test the dependency of initial dropping variables: velocity, height and shoot angle.

To study the relationship between factors and each corresponding part final landing orientation, the Design of Experiment (DOE) method is utilized. A controlled factor design is introduced in this method, and variables in this experiment are assumed to be independent. Variables and each corresponding levels is shown in Table 2.

Table 2: Control Variables and Values in DOE

Control Variable	No. of Levels	Value at different levels
Velocity (m/s)	3	$X_0 = 0.3$, $X_1 = 0.2$, $X_2 = 0.1$
Height (m)	3	$X_0 = 0.4$, $X_1 = 0.3$, $X_2 = 0.2$
Angel (Degree)	3	$X_0 = 90$, $X_1 = 80$, $X_2 = 70$

A total of 630 runs of analyses have been completed to test for the variable dependency of the part final landing orientation. A detailed Design of Experiment representation is shown in Table 3. For each set of initial conditions, 70 runs were conducted. Appendix M. explains the reason of utilizing 70 as the number of runs to get sufficient and accurate results. A detailed FEA analyses results for each variable is shown in Appendix Q.

Table 3: Design of Experiment Representation

	Level used for the Variable		
Total Runs	Velocity (m/s)	Height (m)	Angle (Degree)
70	2	2	0
70	2	2	1
70	2	2	2
70	0	2	0
70	1	2	0
70	2	2	0
70	2	0	0
70	2	1	0
70	2	2	0

Chi square test is used to determine whether there is a significant association between the variable and the part final landing orientation. A detailed chi square test procedure is shown in Appendix P. The chi square test final results are shown in Table 4.

Table 4: Chi Square Test Results for Variable Dependency

Variables	Velocity	Height	Angle
Chi-square Test Value	7.24	9.18	5.43
P-Value	0.3	0.16	0.49

Since all the p-value is greater than the significance level (0.05), there is no significant relationship between varying values of each variable and the part final landing orientations. Therefore, initial dropping velocity, height, and angle will not have effect on the part final landing orientation based on the analysis done on simple model.

c. Applying Computational to a Complex Model

After the computational method was proven with parts of simpler geometry it was then tested against a more complex part. For this a rapid prototype model was used so the results could be compared to physical testing. The model is shown in Fig. 67

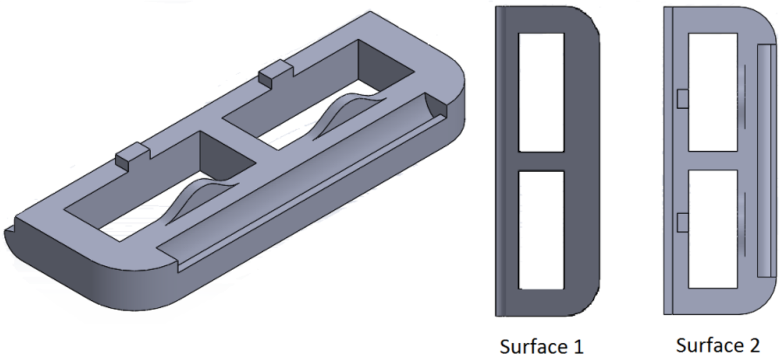


Fig. 67. Simplified model of product, also known as complex model.

After running convergence tests on the part the outcome was a nodal count of 622. Once the part was properly meshed and verified, it was ran through our script with the following initial conditions: a height of 0.2m, an initial velocity of -1m/s, a shoot angle of 90 degrees, and a random orientation. This was set this way in order to ensure that the physical testing conditions could be replicated. The script was set to run for 20 iterations. Even though ideally a convergence run should have been done to figure out how large the sample pool should have been, we did not have time for this considering the computational cost of running these simulations. The main goal from this test was not to predict the parts orientation with the best accuracy that we could achieve but rather to prove that our computational method could be applied to more complex, realistic parts. Fig. 68 shows the comparison between our computational tests and the physical results.

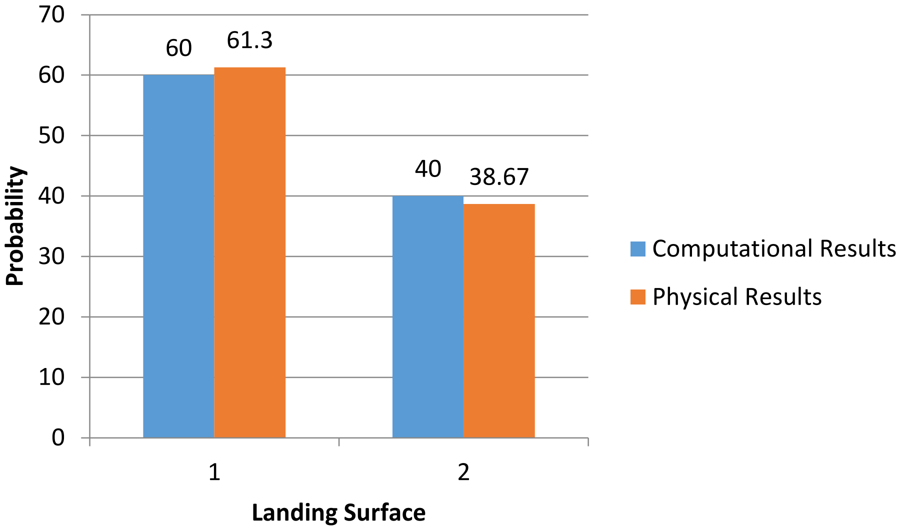


Fig. 68. Computation results vs. physical results for the tests with the complex model.

There was a 2.8% error when comparing the results obtained from the computational method from the results obtained from the physical testing. However only a sample size of 20 runs was put through our script, this was due to computational costs and time restraints. With a sample size of 20 we can conclude that with a 95% confidence level these results are accurate to a confidence interval of 18%. Even though this may not be an ideal case it still shows the script can operate with more complex geometry and obtain results. Further testing can be undergone for future work to sufficiently prove this method works is able to get accurate results with complex geometry.

4.3.1. Validation of Theoretical and Computational Tools with physical testing

When comparing the theoretical method to the computational method, there are two main points that needs to be taken into consideration. The first point is the time and cost of running these methods. Assuming the geometry is already modeled in 3D software the time taken between the two methods varies drastically. Even though the methods may take approximately the same amount of man-hours to set up and gather data form, their run times vary exponentially. The computational method could take upwards of 3-4 days or as little as 1 day to obtain full results. This all depends on the model size, and the computer the simulation is being run on. Even if the model size is relatively small the theoretical method still is much faster at producing the results, almost instantaneously when compared to the computational method.

The next point of comparison is the accuracy of these simulations. If time and computational cost is not a factor, and the most realistic and accurate solution is desired. Then the computational method would be the best choice. However if there is a tight deadline, or simply the appropriate resource's required for the computational method aren't available, then the theoretical method would be the best choice. In terms of accuracy of these methods both Fig. 69 and Fig. 70 show how these how these methods compare in our different simulations.

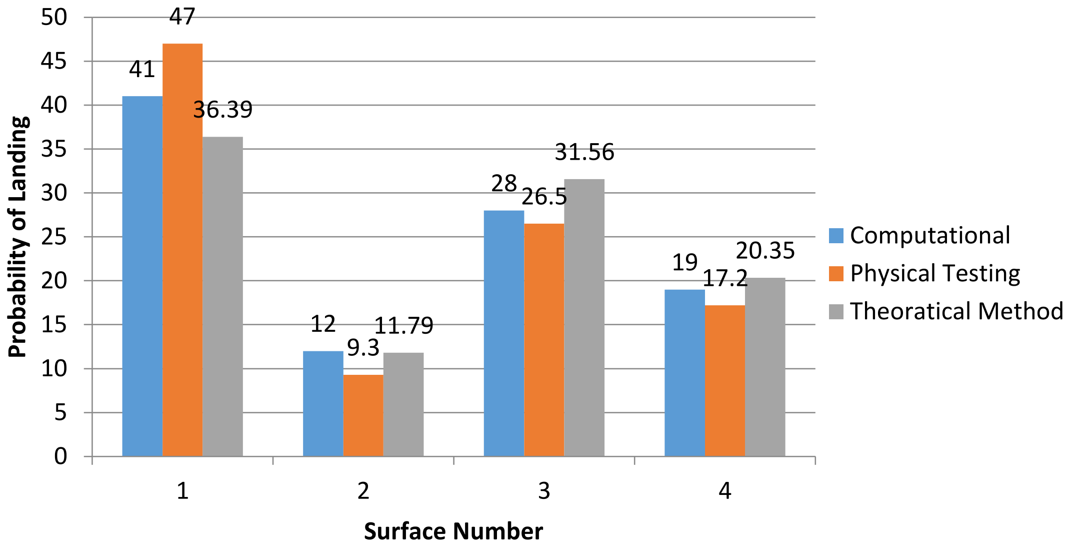


Fig. 69. Overall comparison of methods for simple geometry.

For the part with simple geometry, which can be seen in Fig. 69, the computational error with respect to the physical testing was 14.5%, while the theoretical error was 22.3%. These results are both close to the physical testing. However, the computational method has almost half the error size as the theoretical method. The computational method in this case took around 24 hours to run while the theoretical only took 0.5 hours.

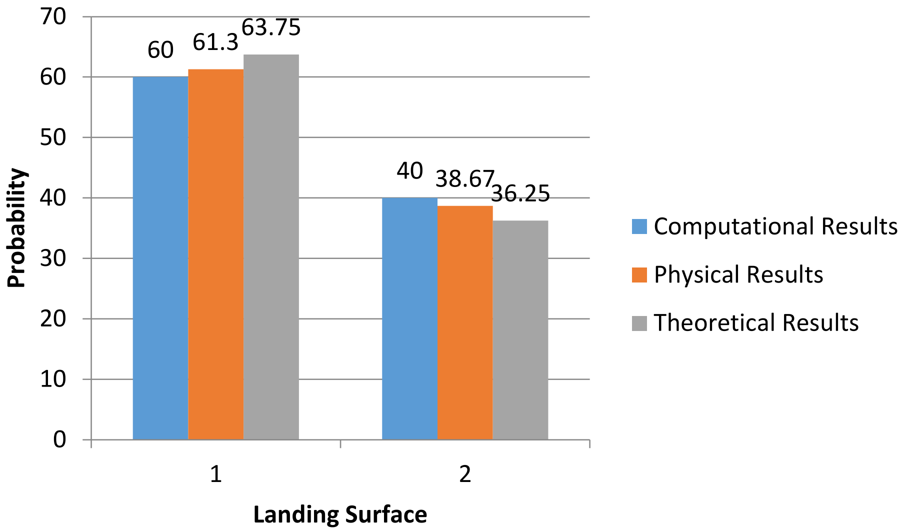


Fig. 70. Overall comparison among methods for the simplified model of the product of interest.

Fig. 70 shows the comparison between the methods when applied to the simplified model of the product of interest as shown in Fig.67, the average percent error for the computational part was 2.8% while the average error for the theoretical method was 5.8%. While the computational method had approximately half the amount of error it also took a lot longer to run. The total computational simulation for this test took around 36 hours to run while the theoretical method only took around one and a half hours.

Although the runtimes for the computational method are not quantifiable since the simulations were ran using a shared and limited server resources. In addition to this the runtime of the computational method depends highly on the computers' CPU, with more and better CPUs the runtime will decrease. However the time of the computational results will still take exponentially longer than the theoretical results. So this becomes a balance of accuracy and time and which one is preferred in the current scenario.

5. Conclusions

5.1. Theoretical Method

The theoretical portion of this project provides the sponsor with a simple tool to predict the percentage occurrences of final orientations of parts being dropped. The developments are based on the surface areas of the landing surfaces and their distances from the center of gravity. More specifically the Stability Method was used as the based theory for this portion of the project. The original Stability Method was developed to be used with more complex parts, more specifically, the product of interest. The results obtained from the theoretical model were validated with physical testing and it can be said with 90% confidence that the difference in the set of data obtained from applying the New Modified Stability Method and the data obtained from physical testing is likely due to chance. The theoretical portion of this project is also able to provide the sponsors with recommendations on where the center of gravity should be located to obtain the desired part orientation ratio. This makes it easier for a designer to manipulate the dimensions and material properties of a part so that it has a desired landing orientation and could potentially increase the efficiency of assembly.

5.2. Computational Method

Overall, the computational method provides the sponsor with a tool to first simulate the randomized part dropping process in ABAQUS with a given CAD model, and then interpret the finite element analysis results in MATLAB to obtain the probability distribution of part final landing orientations. The tool also enables the designer to simulate the part dropping under various initial conditions (velocity, height and shoot angle) and allows the part to have multiple material properties, thus the designer can manipulate materials of part or initial conditions of dropping to obtain a desired part final landing orientation.

5.3. Overall

As stated before, both methods have proved to be a viable way to predict the final orientation of the parts. The Theoretical Method, while less accurate than the Computational Method is less time consuming and have less computational cost. Whereas the Computational Method, although more accurate, undergoes a much larger computation cost, with respects to required resources and time. So in the end the question is whether accuracy or time is preferred for testing. This would be something the sponsor or user would have to weigh before deciding when to use each method.

6. Future Work/ Development

6.1. Theoretical

The future developments of the theoretical portion of this project can be divided into two main categories.

The theory can be further validated by conducting tests on more objects with varying geometries as to identify any changes in surface area or distance from center of gravity considerations. In terms of mathematical modeling, the equation itself can be developed to reflect the findings in this paper. In other words, the equation can be modified to reflect the different surface area calculations and considerations.

The script and the user interface can also be developed further. Future developments can include the automation of the calculation of the variables, meaning the process of finding the surface areas and their respective distances from the center of gravity can be fully automated to decrease the amount of user input.

6.2. Computational

Due to time constraints and limited computing resources, there still is plenty of work that can be done in terms of computational data. The project has successfully modeled and predicted results of simple parts with simple geometry and has been verified to work with more complex models. However the current method of using surfaces to define the landing orientation can become cumbersome when dealing with complex geometry due to the vast number of surfaces. It would be beneficial to find another method to get output data, which can be used to get the parts final orientation. This way no surfaces have to actually be defined, as it would then be possible just to use whole model contact instead of surface-to-surface contact. In addition to this we were not able to look into how various material properties affect the outcome of these results, along with combining multiple material properties in a single part. Additional test are also needed to further prove our hypothesis for variable dependency.

6.3. Experimental

Future work on proving hypothesizes with regards to variables like the velocity and shoot angle are needed. Although we were able to run simulations to simulate these results nothing has been acquired to back up our simulations at this point. Because of this it can only hypothesize what may or may not have an effect on the parts final orientation. For this a testing rig that will be able to control all of our variables, height, initial velocity, initial orientation, and finally feed angle is required. Once a testing rig is constructed the hypotheses can then be tested and backed up.

7. References

Assembly Line. (2003). Retrieved from Encyclopedia.com: http://www.encyclopedia.com/doc/1G2-3401800294.html

Berkowitz, D. R., & Canny, J. (1996). Designing Parts Feeders Using Dynamic Simulation. *Interinational Conference on Robotics and Automation*, 1127-1132.

Berkowitz, D. R., & Canny, J. (1996). Designing Parts Feeders Using Dynamic Simulation. *International Conference on Robotics and Automation* , 1127-1123.

Dassault Systems Simulia Corp. (2012, Feb 13). *Abaqus 6.12 Online Documentation*. Retrieved Apr 15, 2015, from http://www.maths.cam.ac.uk/computing/software/abaqus_docs/docs/v6.12/books/hhp/default.htm

Goss, J. L. (2010). *About education*. Retrieved from http://history1900s.about.com/od/1910s/a/Ford--Assembly-Line.htm

Halt & Hass. (n.d.). Retrieved from http://www.halthass.co.nz/reliability-services/environmental-testing-category/product-drop-testing/

Jaksic, I. N., & Maul, P. G. (2001). Development of a model for part reorientation in vibratory bowl feeders with active air jet tooling. *Robotics and Computer Integrated Manufacturing*, 145 - 149.

Light, C. (2008). *Tutorial: Pearson's Chi-square Test for Independence*. Retrieved 2014, from University of Pennsylvania: http://www.ling.upenn.edu/~clight/chisquared.htm

Maul, G., & Thomas, B. M. (1997). A Systems Model and Simulation of the Vibratiory Bowl Feeder. *Journal of Manufacturing Systems*, 309-314.

Ngoi, B., Lim, L., & Ee, T. (1993). Analysis of Natural Resting Aspects of Parts in a Vibratory Bowl Feeder - Validation of "Drop Test" . *The International Journal of Advanced Manufacturing Technology*, 300-309.

Silversides, R., Dai, J. S., & Seneviratne, L. (2004). Force Analysis of a Vibratory Bowl Feeder for Automatic Assembly. *Kings College London*.

Sprovieri, J. (2001, Jul 01). *Parts Feeding: The Black Art of Tooling Feeder Bowls*. Retrieved from Assembly : http://www.assemblymag.com/articles/84051-parts-feeding-the-black-art-of-tooling-feeder-bowls

Suresh, M., Jagadeesh, K., & Varthanan, A. P. (2013). Determining the natural resting orientation of a part using drop test and theoretical methods. *Journal of Manufacturing Systems*, 220 -227.

Udhayakumar, S., Mohanram, P., Keerthi Anand, P., & Srinivasan, R. (2013). Determining the most probable natural resting orientation of sector shaped parts. *Assembly Automation*, 29-37.

Weisstein, E. W. (n.d.). *Solid Angle*. Retrieved from Wolfram MathWorld: http://mathworld.wolfram.com/SolidAngle.html

Appendix

Appendix A. Centroid Solid Angle Method Results vs. Stability Method Results

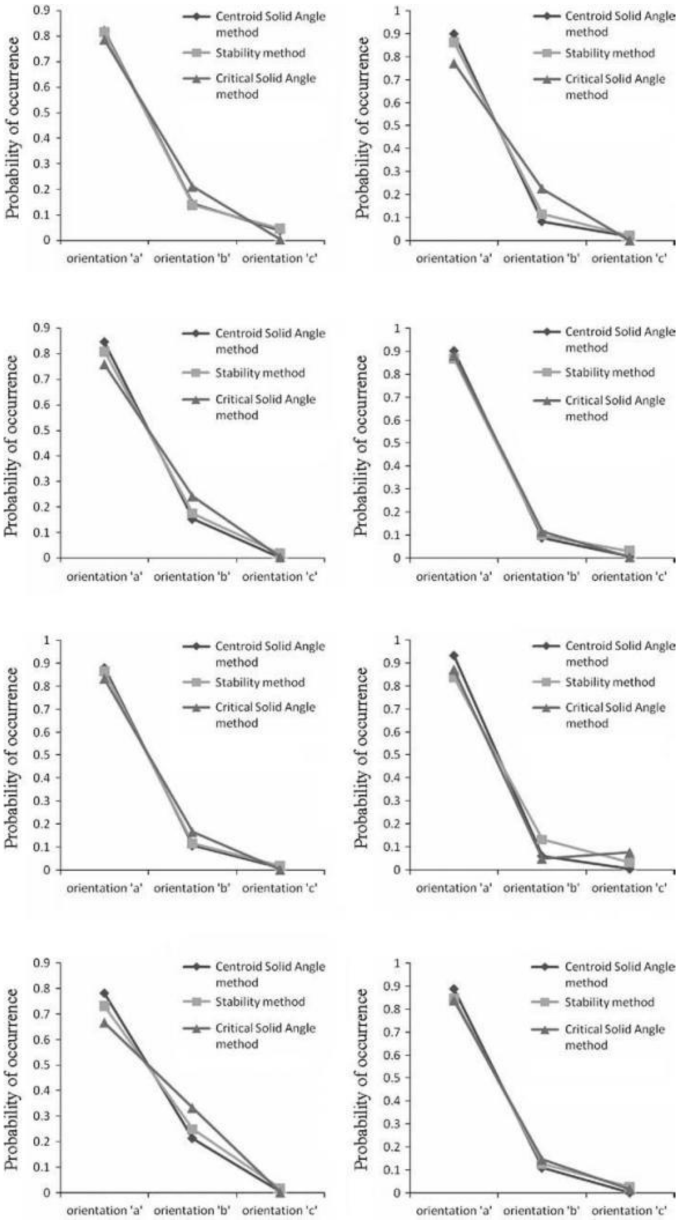


Fig. 71. Results obtained from the Centroid Solid Angle Method versus the Stability Method of eight different objects.

Appendix B. Values of Variables considered

Rectangular Prism: shown in Fig.6

Table 5. Values of area of contacting surface and distance of the surface from the center of gravity for a rectangular prism with aspect ratio (y/x) of 0.56

y = 10.00 , x = 17.86 , aspect ratio y/x = 0.56				
	Area of Contacting Surface (mm2)	Number of Surfaces	Distance of surface from center of gravity (mm)	Percentage of Occurrence
Surface a	159.445	2	5	38.94
Surface b	89.29	2	8.93	12.21
Surface c	178.57	2	4.47	48.84

Object 1: shown in Fig. 8.

Table 6. Values of surface area and distance from the surface from the center of gravity of Object 1 for Original Stability Method

Object 1				
	Area of Contacting Surface (mm2)	Number of Surfaces	Distance of surface from center of gravity (mm)	Percentage of Occurrence
Surface 1	32.22	1	23.46	0.42
Surface 2	4461.06	1	20	57.84
Surface 3	1256.64	1	12.04	31.74

Table 7. Values of surface area and distance from the surface from the center of gravity of Object 1 for Modified Stability Method

Object 1				
	Area of Contacting Surface (mm2)	Number of Surfaces	Distance of surface from center of gravity (mm)	Percentage of Occurrence
Surface 1	1256.64	1	23.46	14.06
Surface 2	4461.06	1	20.00	58.55
Surface 3	1256.64	1	12.04	27.39

Object 2: shown in Fig. 11.

Table 8. Values of surface area and distance from the surface from the center of gravity of Object 1 for Original Stability Method

Object 2				
	Area of Contacting Surface (mm2)	Number of Surfaces	Distance of surface from center of gravity (mm)	Percentage of Occurrence
Surface 1	215.78	2	17.00	12.54
Surface 2	750.82	1	5.38	68.91
Surface 3	29.58	1	6.68	2.19
Surface 4	23.12	1	9.64	1.18
Surface 5	300.00	1	9.76	15.18

Table 9. Values of surface area and distance from the surface from the center of gravity of Object 2 for Modified Stability Method

Object 2				
	Area of Contacting Surface (mm2)	Number of Surfaces	Distance of surface from center of gravity (mm)	Percentage of Occurrence
Surface 1	215.78	2	17.00	7.08
Surface 2	952.00	1	5.38	49.38
Surface 3	746.75	1	6.68	31.20
Surface 4	129.10	1	9.64	3.74
Surface 5	300.00	1	9.76	8.6

Object 3: shown in Fig. 25.

Table 10. Values of surface area and distance from the surface from the center of gravity of Object 3 for New Modified Stability Method

Object 3				
	Area of Contacting Surface (mm2)	Number of Surfaces	Distance of surface from center of gravity (mm)	Percentage of Occurrence (approach 1)
Surface 1	2732.30	2	9.50	75.18
Surface 2	1765.18	1	16.81	13.73
Surface 3	973.248	1	10.33	8.72
Surface 3.1	909.73	1	28.23	
Surface 4	687.90	1	37.89	2.37

Object 4: shown in Fig. 72.

Table 11. Values of surface area and distance from the surface from the center of gravity of Object 4 for Modified Stability Method

Object 4				
	Area of Contacting Surface (mm2)	Number of Surfaces	Distance of surface from center of gravity (mm)	Percentage of Occurrence (approach 1)
Surface 1	504.63	2	9.25	
Surface 2	412.55	1	11.68	
Surface 3	625.18	1	8.23	
Surface 3.1	153.43	1	14.57	
Surface 4	589.394	1	9.72	

Simplified Product: shown in Fig. 14.

Table 12. Values of surface area and distance from the surface from the center of gravity of the simplified product for Original Stability Method

Simplified Product				
	Area of Contacting Surface (mm2)	Number of Surfaces	Distance of surface from center of gravity (mm)	Percentage of Occurrence
Surface 1	321.2	1	1.91	98.36
Surface 2	9.98	1	3.57	1.64

Table 13. Values of surface area and distance from the surface from the center of gravity of Object 4 for Modified Stability Method

Simplified Product				
	Area of Contacting Surface (mm2)	Number of Surfaces	Distance of surface from center of gravity (mm)	Percentage of Occurrence
Surface 1	525.95	1	1.94	89.15
Surface 2	117.8	1	3.57	10.85

Table 14. Values of surface area and distance from the surface from the center of gravity of Object 4 for Modified Stability Method

Simplified Product				
	Area of Contacting Surface (mm2)	Number of Surfaces	Distance of surface from center of gravity (mm)	Percentage of Occurrence (approach 1, approach 2
Surface 1	525.95	1	1.94	63.75 ,77.51
Surface 2	117.8	1	3.57	36.25, 22.49
Surface 2.1	69.95	2	10.04	
Surface 2.2	109.2	1	4.78	
Surface 2.3	44.19	2	9.92	
Surface 2.4	95.03	1	7.02	

Appendix C. Physical Testing Results for Rectangular Prism as described in Section 3.2.3.

Table 15. Physical Testing Results of a Rectangular Prism on a Vibratory Bowl with a Hard Surface obtained from (Ngoi, Lim, & Ee, 1993)

Amplitude Level: 0.07mm

y/x	Total no.of components	No. of components with aspect a					
		Test 1	Test 2	Test 3	Test 4	Test 5	Average
0.30	100	80	79	81	83	79	80.40
0.40	100	67	69	69	79	69	68.60
0.56	100	42	43	44	69	45	43.00
0.72	100	26	23	28	45	22	25.00
0.84	100	13	16	17	22	19	15.00
1.00	100	10	8	6	19	7	7.60
1.26	100	3	3	5	2	4	3.40
1.67	100	0	1	0	0	1	0.40

y/x	Total no.of components	No. of components with aspect b					
		Test 1	Test 2	Test 3	Test 4	Test 5	Average
0.30	100	2	1	2	1	1	1.40
0.40	100	2	3	1	2	2	2.00
0.56	100	3	4	6	8	6	5.40
0.72	100	7	8	5	8	8	7.20
0.84	100	4	5	4	6	7	5.20
1.00	100	10	8	6	7	7	7.60
1.26	100	11	6	12	8	7	8.80
1.67	100	11	9	11	12	10	10.30

	Total no.of	No. of components with aspect c					
y/x	components	Test 1	Test 2	Test 3	Test 4	Test 5	Average
0.30	100	18	20	17	16	20	18.20
0.40	100	31	28	30	29	29	29.40
0.56	100	55	53	50	51	49	51.60
0.72	100	67	69	67	66	70	67.80
0.84	100	83	79	79	84	74	79.80
1.00	100	80	84	88	86	86	84.80
1.26	100	86	91	83	90	89	87.80
1.67	100	89	90	89	88	89	89.00

Appendix D. Sample Size Calculation

Determining the sample size of an experiment is extremely important. If the sample size is too large, then more time and resources is being put into preforming the physical experiments. However, if the sample size is too small, then the results may not be accurate. The minimum sample size needed to estimate the determine the probabilities of the landing surfaces of different types of objects are determined by a number of factors, the number of landing surfaces, the confidence interval and the confidence level. The equation to calculate sample size is

$$n = \frac{p(1-p)(z_{\frac{1+L}{2}}^{2})}{d^2}, \tag{D1}$$

where *n* is the sample size, *p* is the probability *z* is the z value, *d* is the confidence interval and *l* is the confidence level . The confidence interval and the confidence level is specified by the team, to be 0.9 and 0.05. The corresponding z value is 1.645. The probability is determined from the number of possible landing surfaces and found using the equation:

$$p = \frac{1}{Number\ of\ Posisble\ Surfaces} \tag{D2}$$

The value of the sample size is determined for each part so that the appropriate number of experiments can be performed. Table 3 shows the minimum sample size required for each object discussed in the paper.

Table 16. Number of landing surfaces and minimum sample size required for physical testing. Minimum sample size value rounded to the next whole number.

Object	Number of Unique Landing Surfaces	Minimum Sample Size
Object 1	3	241
Object 2	4	203
Object 3	4	203
Object 4	4	203
Simplified Product	2	271

Appendix E. Chi Square Test in Theoretical Method

The Chi Square Test was used to determine if there was a significant association between the results from the theoretical and physical testing. It is used to evaluate whether the amount of difference in the sets of values for from the physical testing and the theoretical results is expected or due to chance. However, the test does not give any details about the relationship between the two sets of data. The Chi Square equation is

$$X^2 = \sum \frac{(O-E)^2}{E} \tag{E1}$$

where X^2 is the Chi Square value, O is the observed value and E is the expected value.

The Chi Square test value is compared to the p-value. The p-value is based on the degrees of freedom, the specified significance level and the Chi-square test value. The smaller the p-value, the larger the statistical significance. This means that the larger the p-value, the greater the confidence that the difference between the two sets of data are not statistically significance (Light, 2008). The p-value can be obtained using the function '=CHITEST (actual_range, expected_range)' in Microsoft Excel. Table 17 shows the p-values and Chi-square test values for all the parts investigated in the paper.

Table 17. Chi Square Test Results where Physical Testing = PT, Original Stability Method = OSM, Modified Stability Method = MSM, New Modified Stability Method = NMSM

Object	Data set 1	Data set 2	P-Value	Chi Square Test Value	Significance at p<10
Rectangular Cuboid Prism	PT	OSM	0.2234	4.3774	Difference is not significant
Object 1	PT	OSM	< 0.00001	321.7574	Difference is significant
Object 1	PT	MSM	0.4347	1.6661	Difference is not significant
Object 2	PT	OSM	<0.00001	536.8054	Difference is significant
Object 2	PT	MSM	0.8507	1.3288	Difference is not significant
Object 3	PT	NMSM	0.9220	0.5774	Difference is not Significant
Object 4	PT	NMSM	0.1768	4.8647	Difference is not significant
Simplified Product	PT	OSM	< 0.00001	849.8981	Difference is significant
Simplified Product	PT	MSM	< 0.00001	179.9943	Difference is significant
Simplified Product	PT	NMSM, approach 1	0.6151	0.2527	Difference is not significant
Simplified Product	PT	NMSM, approach 2	0.6152	4.2659	Difference is significant
Modified Simplified Product	PT	NMSM, approach 1	0.8060	0.0603	Difference is not significant

Appendix F. Data from Physical Testing

Object 2: shown in Fig. 11.

Table 18. Physical Testing Data for Object 2

Surface	Number of drops	Percentage occurrence
Surface 1	129	43.00
Surface 2	16	5.33
Surface 3	155	51.67
Total Number of drops	300	

Object 3: shown in Fig. 25.

Table 19. Physical Testing Data for Object 3

Surface	Number of drops	Percentage occurrence
Surface 1	18	5.81
Surface 2	142	45.81
Surface 3	111	35.81
Surface 4	14	4.52
Surface 5	25	8.06
Total Number of drops	310	

Object 4: shown in Fig. 72.

Table 20. Physical Testing Data for Object 4

Surface	Number of drops	Percentage occurrence
Surface 1	235	76.55
Surface 2	37	12.05
Surface 3	30	9.77
Surface 4	5	1.63
Total Number of drops	307	

Simplified Product: shown in Fig. 14

Table 21. Physical Testing Data for Simplified Product

Surface	Number of drops	Percentage occurrence
Surface 1	414	61.33
Surface 2	261	38.67
Total Number of drops	675	

Modified Simplified Product shown in Fig. 29

Table 22. Physical Testing Data for Modified Simplified Product

Surface	Number of drops	Percentage occurrence
Surface 1	248	52.77
Surface 2	222	47.23
Total Number of drops	470	

Appendix G. Percentage Error Calculations

The equation for percentage error calculations is

$$Percentage\ Error = \frac{|Experimental\ Value - Theoretical\ Value|}{Theoretical\ Value}\ x\ 100 \qquad \text{(G1)}$$

The data obtained from physical testing is considered the theoretical value in the equation since the data from the drop tests are considered the real expected values. The two sets of data obtained from Approach 1 and Approach 2 of the New Modified Stability Method are considered the experimental value in the equation. Table 23 shows the percentage errors for the simplified product obtained from the calculations.

Table 23. Percentage Error for Simplified Product

	Percentage Error	
Surface	Physical Data vs Approach 1	Physical Data vs Approach 2
Surface 1	3.94	26.38
Surface 2	6.25	41.84
Average	5.10	34.11

Appendix H. Application of New Modified Stability Method to Object 4

An isometric view and dimensions of Object 4 can be found in Fig. 72 and the different landing surfaces are found in Fig. 73.

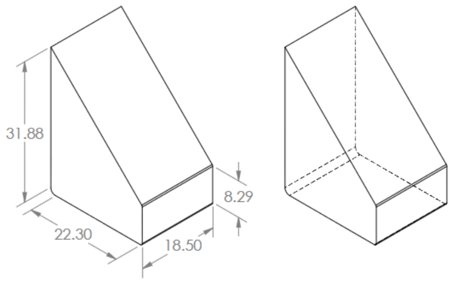


Fig. 72. Object 4, a part used to prove the concept of the New Modified Theory. (a) Isometric view of Object 4 with dimensions, measured in millimeters; (b) Isometric view of Object 4 with hidden lines.

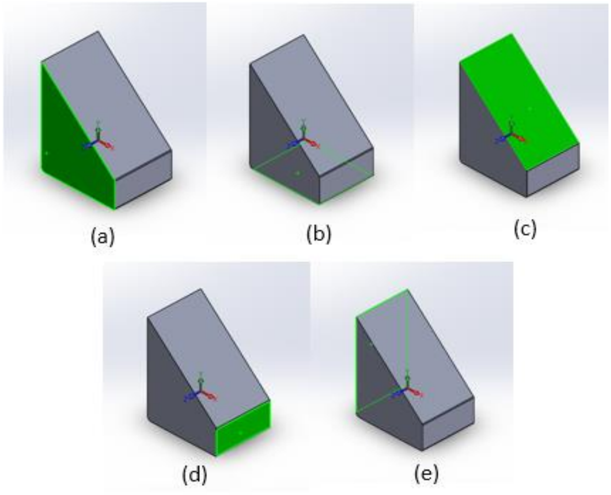


Fig. 73. Different landing surfaces of Object 4 considered in the Modified Stability Method, area highlighted in green (a) Landing Surface 1; (b) Landing surface 2; (c) Landing surface 3; (d) Landing Surface 3.1; (e) Landing Surface 4.

The data obtained from applying the New Modified Stability Method is compared to the data obtained from physical testing in Fig. 74. The P-value and the Chi-square test value is 0.1768 and 4.8647 respectively. Thus, it can be said with 90% confidence that the difference in percentages occurrence from the data obtained is likely due to chance.

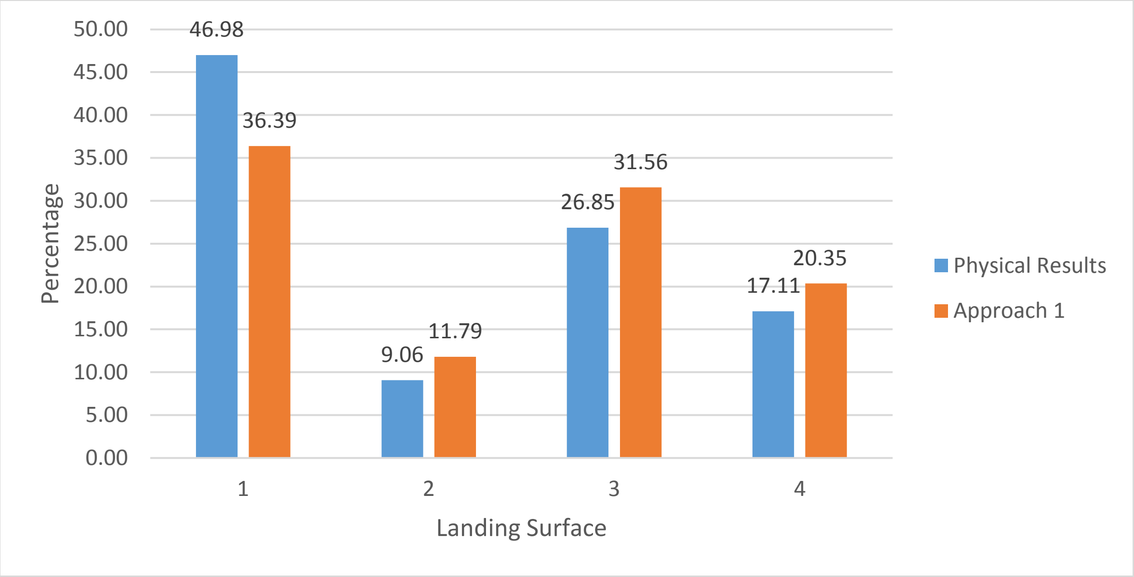
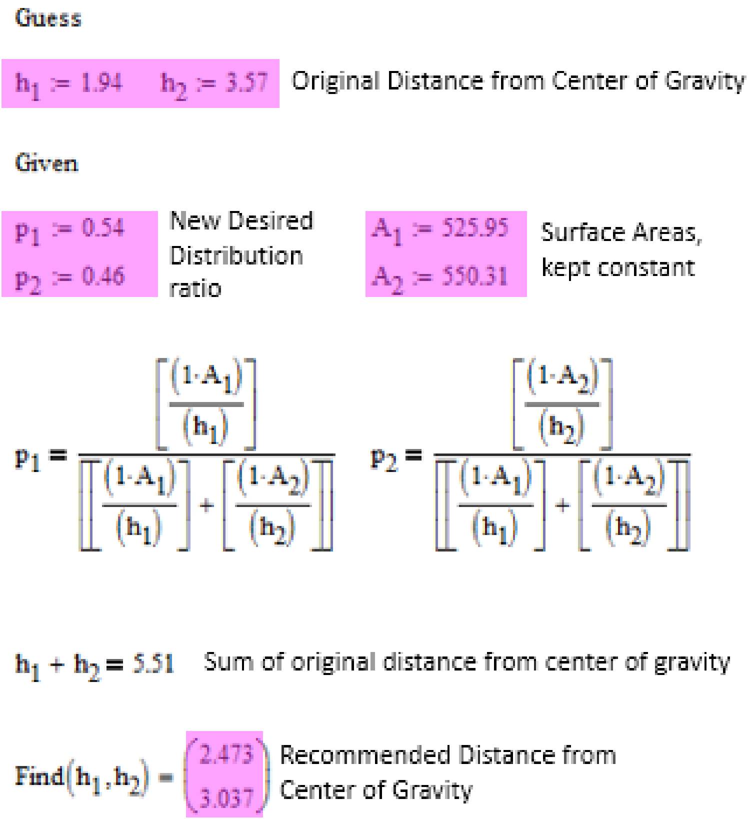


Fig. 74. Results comparison between physical testing, original stability method and modified stability method results for Simplified Product

Appendix I. Calculation Example from Theoretical Method



Guess

$h_1 := 1.94 \quad h_2 := 3.57$ Original Distance from Center of Gravity

Given

$p_1 := 0.54$
$p_2 := 0.46$ New Desired Distribution ratio

$A_1 := 525.95$
$A_2 := 550.31$ Surface Areas, kept constant

$$p_1 = \frac{\left[\frac{(1 \cdot A_1)}{(h_1)}\right]}{\left[\left[\frac{(1 \cdot A_1)}{(h_1)}\right] + \left[\frac{(1 \cdot A_2)}{(h_2)}\right]\right]} \qquad p_2 = \frac{\left[\frac{(1 \cdot A_2)}{(h_2)}\right]}{\left[\left[\frac{(1 \cdot A_1)}{(h_1)}\right] + \left[\frac{(1 \cdot A_2)}{(h_2)}\right]\right]}$$

$h_1 + h_2 = 5.51$ Sum of original distance from center of gravity

$\mathrm{Find}(h_1, h_2) = \begin{pmatrix} 2.473 \\ 3.037 \end{pmatrix}$ Recommended Distance from Center of Gravity

Figure 75. MATHCAD computation recommending new distance from center of gravity for the part of interest.

Appendix J. User Manual for Theory based tool

Prerequisites

Software

- SolidWorks or any CAD program with surface area and distance measurement tool. This manual will explain the process involved in using the program SolidWorks to find the surface area and the distance from the COG, but in general any CAD program that is capable of doing the same measurement can be used.
- MATLAB 8.1 (R2013a) or higher for computation of the percentage occurrences and the recommendations

Others

- CAD model of the part

Instructions

1. Open up the CAD model in SolidWorks 2011 (or later)
2. Displaying the Center of Gravity as a Reference Geometry
    a. Go to ‘Insert’ > ‘Reference Geometry’ > ‘Center of mass’, as shown in Fig. 76. A black and white sphere should appear in the part.

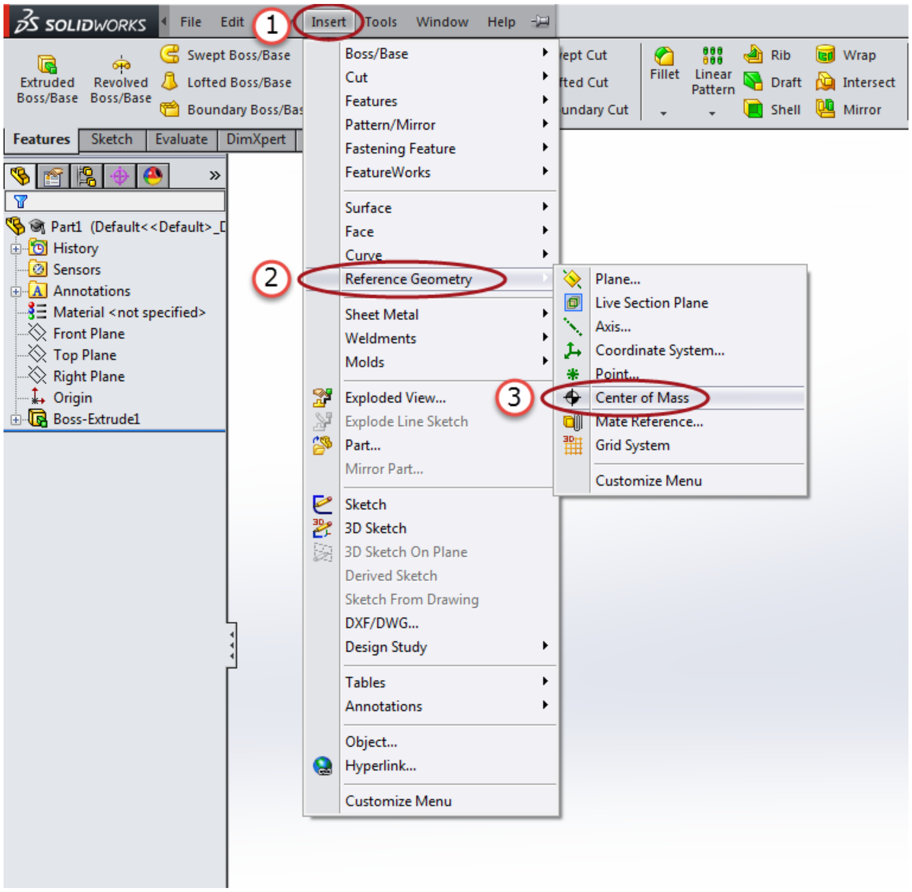


Fig. 76. Steps to display the COG as reference geometry

3. Activating the Measurement tool

   a. Go to ‘Tools’ > ‘Measure’, as shown in Fig. 77. A box similar to Fig. 78 should appear.

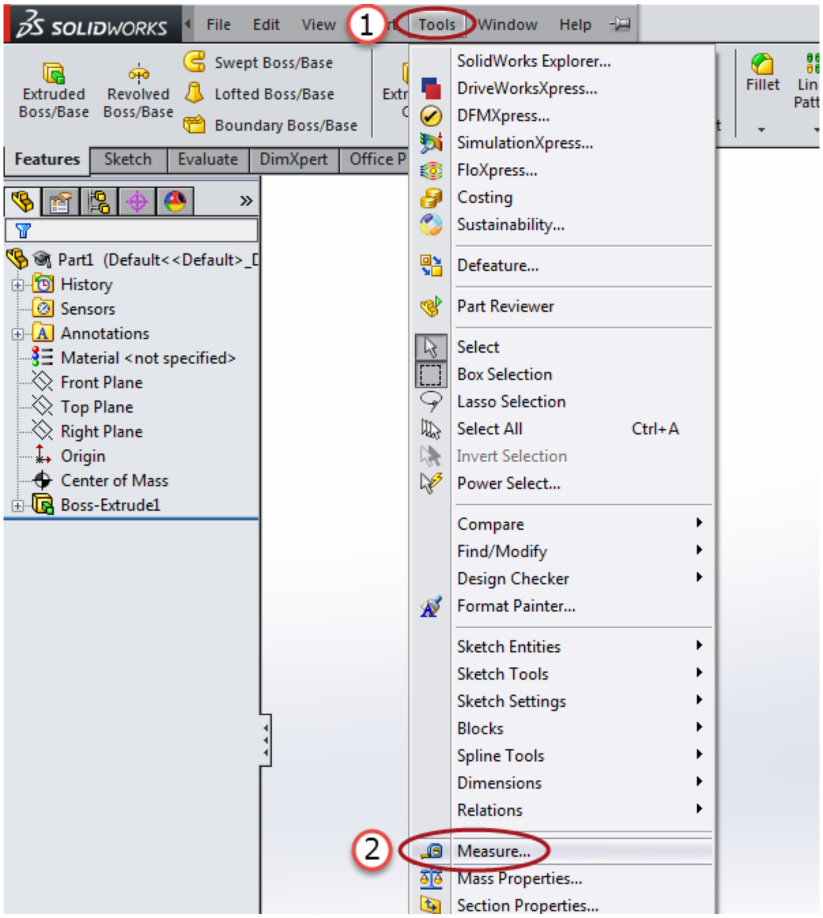


Fig. 77. Activating Measurement Tool in SolidWorks

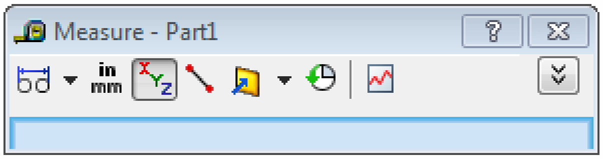


Fig. 78. Measurement Tool

4. Surface Area Measurement

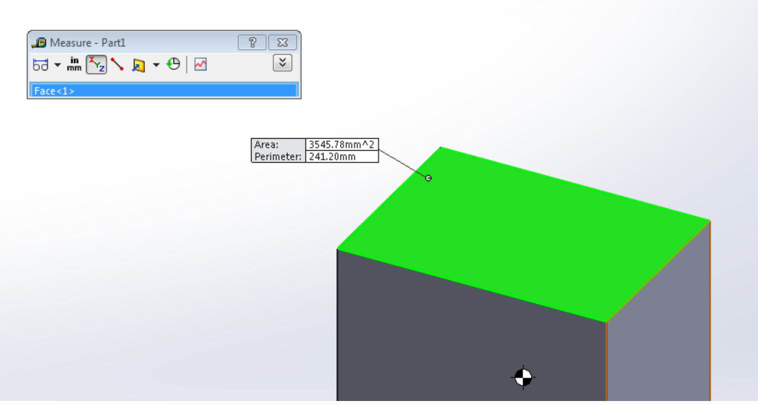
    a. Click on the surface that needs to be calculated while the Measurement Tool is active. The surface should become highlighted and the value of the surface area should appear as shown in Fig. 79.

    b. Record the values of each landing surface, a detailed explanation of what surfaces to consider for different geometries can be found in in the General Rules section. Proceed to the next step if a construction of a plane is needed, if not proceed to Step 5.

    c. Construction of planes

        i. Go to ‘Insert’ > ‘3D Sketch’, as shown in Fig. 80.

        ii. Construct the plane by first creating points on the part then using the line tool to connect the points to create the plane, as shown in Fig. 81.



Face<1>

Area:	3545.78mm^2
Perimeter:	241.20mm

Fig. 79. Surface Area Measurement in SolidWorks

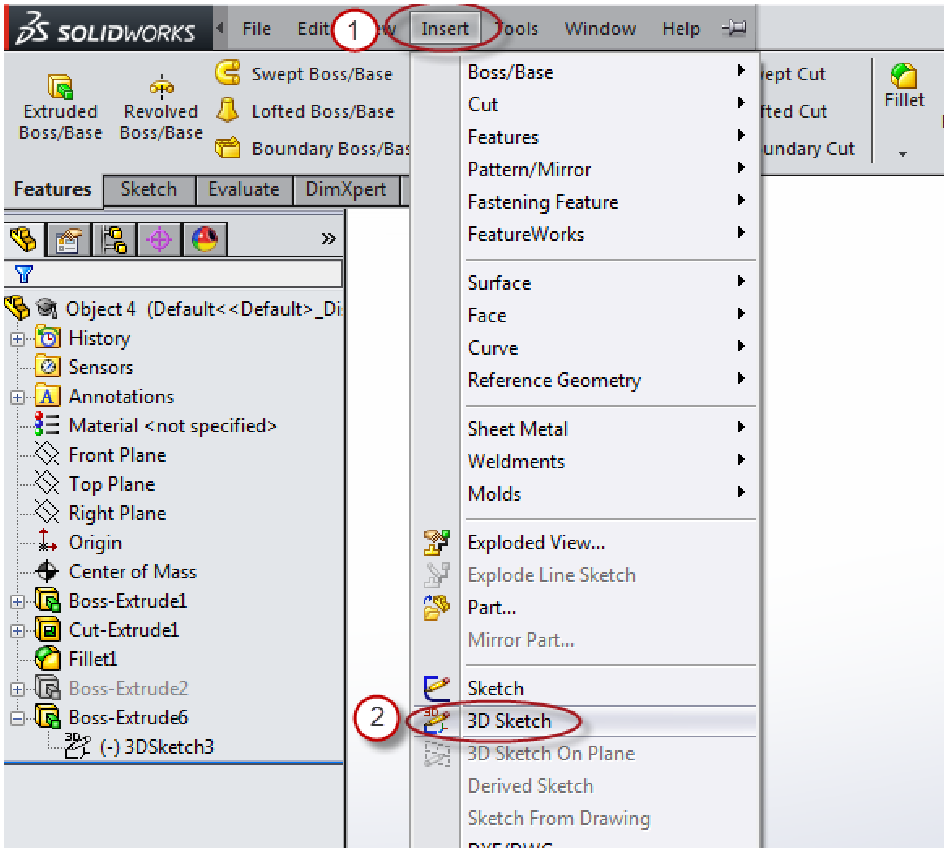


Fig. 80. Insert 3D Sketch

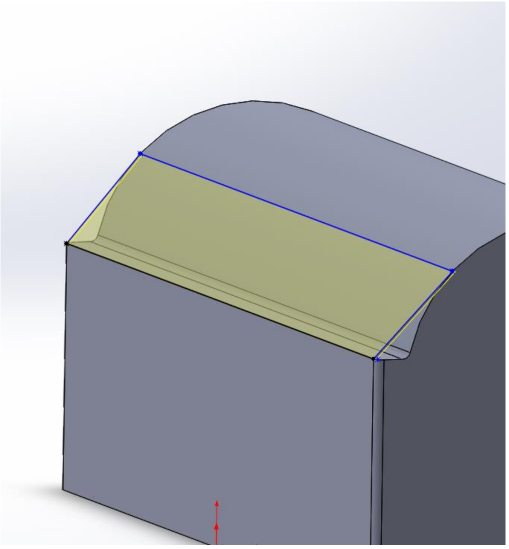


Fig. 81. Plane construction example

5. Distance from Center of Gravity Measurement

   a. Click on the COG, hold down the ‘Ctrl’ button on the keyboard and select the surface while the Measurement Tool is active. A line should form between the COG and the surface and the distance between the two points should be displayed as shown in Fig. 82.

   b. Record the values the distance from the landing surfaces from the COG, a detailed explanation of what COG value to consider can be found in in the General Rules section.

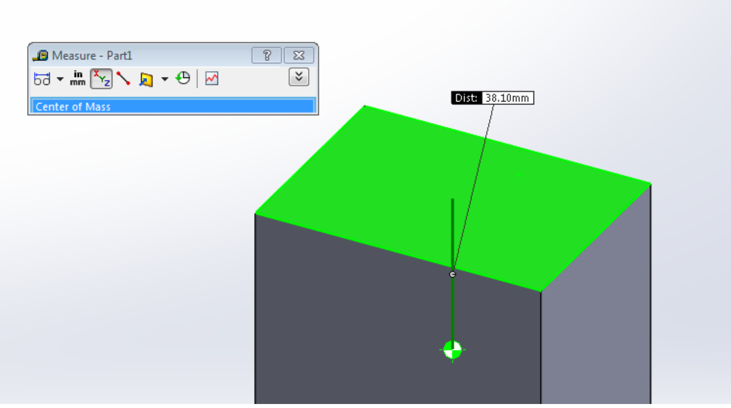


Fig. 82. Distance from COG measurement using SolidWorks

6. Finding the percentage occurrence of each surface using MATLAB

   a. Launch MATLAB

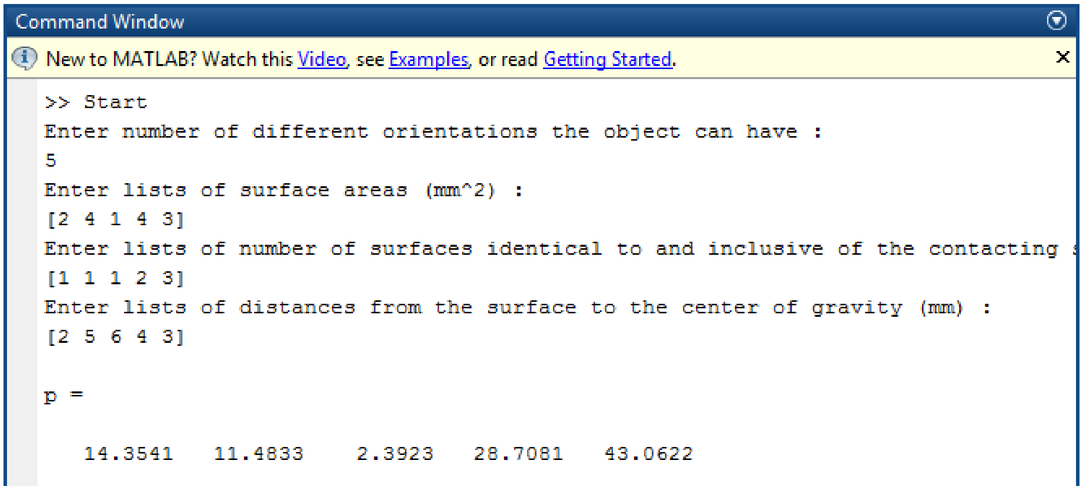
   b. Type “Start ” into the command window.

   c. Manually input the values previously obtained from SolidWorks and from observation. Press Enter on the keyboard to move onto the next stage. The questions and input variable format required are shown in Table 24. An example of the interface is shown in Fig. 83.

d. A histogram showing the percentage occurrences should appear. An example is shown in Fig. 84.

Table 24. Question prompt and input variable format for theoretical tool

Question Prompt	Variable format
Enter number of different orientations the object can have	Positive integer, n
Enter lists of surface areas (mm^2)	$[A_1\, A_2\, A_3\ \ldots\ A_n\,]$
Enter lists of number of surfaces identical to and inclusive of the contacting surface area	$[n_1\, n_2\, n_3\ \ldots\ n_n\,]$
Enter lists of distances from the surface to the center of gravity (mm)	$[y_1\, y_2\, y_3\ \ldots\ y_n\,]$


Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
×

```
>> Start
Enter number of different orientations the object can have :
5
Enter lists of surface areas (mm^2) :
[2 4 1 4 3]
Enter lists of number of surfaces identical to and inclusive of the contacting s
[1 1 1 2 3]
Enter lists of distances from the surface to the center of gravity (mm) :
[2 5 6 4 3]

p =

   14.3541   11.4833    2.3923   28.7081   43.0622
```

Fig. 83. Example of MATLAB interface.

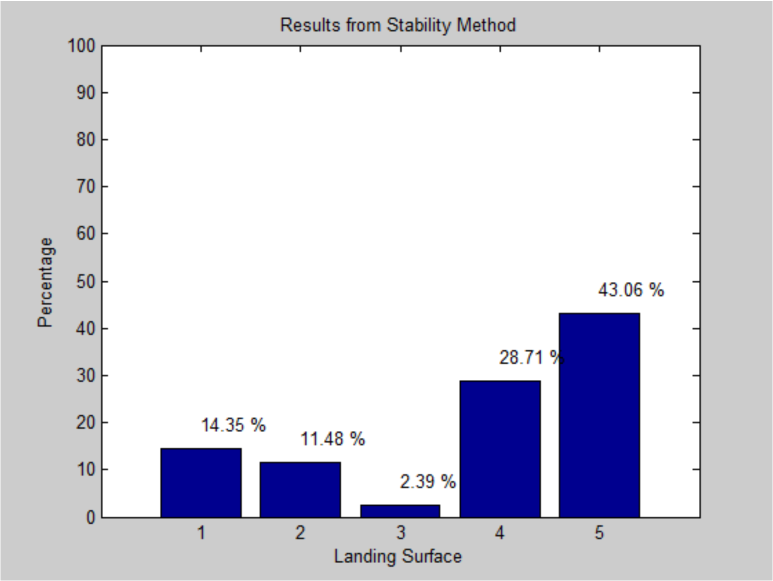


Fig. 84. Example of Histogram showing the results obtained

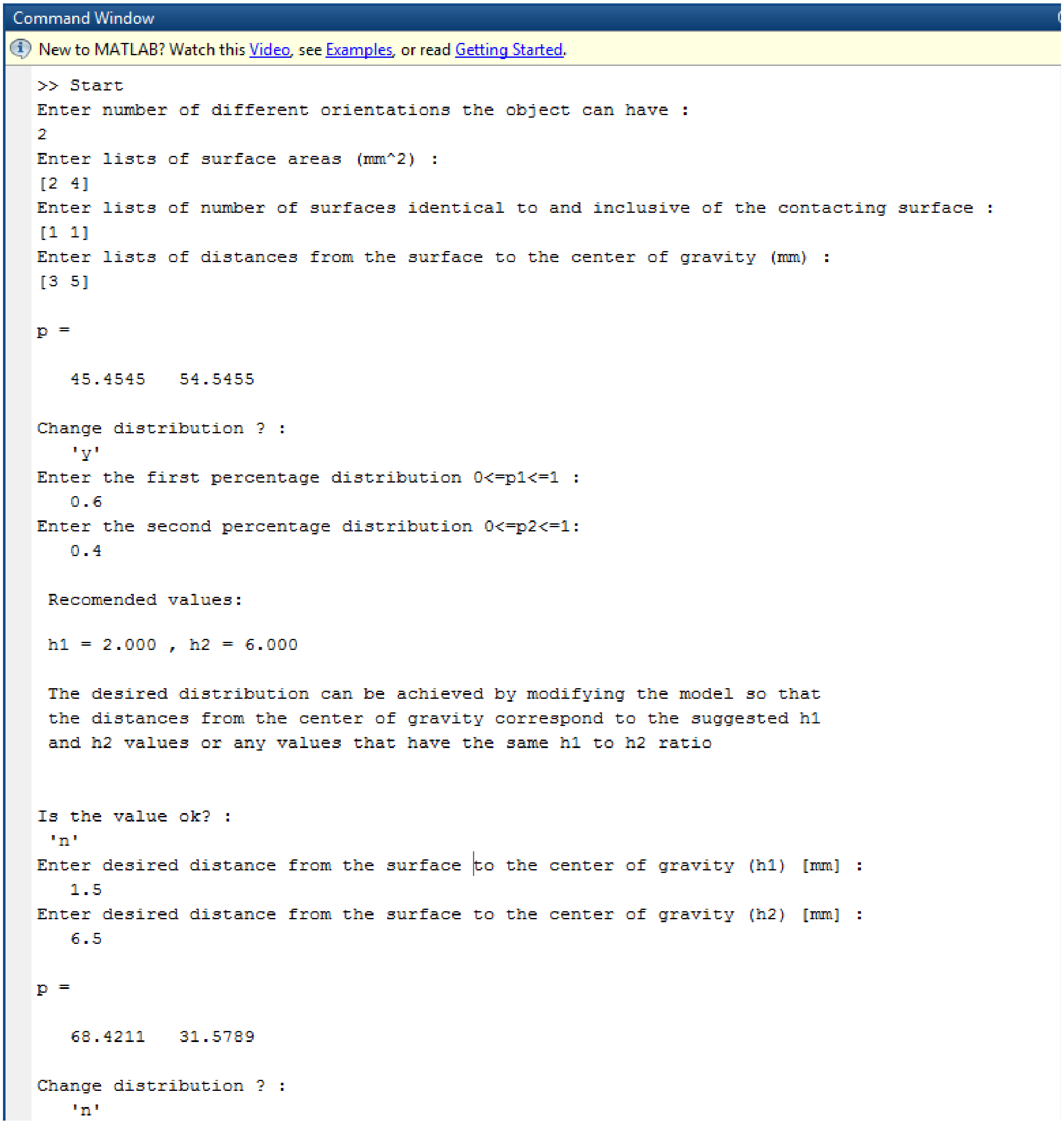
7. Changing the percentage distribution: this portion of the script can only be used with **objects that can be simplified into a rectangular prim and has two stable landing orientations such as the product of interest.** An example of the interface showing the steps can be found in Fig. 85.
   a. The prompt "Change distribution?" will appear on the command window. If change in percentage distribution is desired, type " 'y' " , if not then type " 'n' "and the program will come to an end. If yes, then proceed to the next step.
   b. The prompt will ask for the desired percentage distribution, P_1 then P_2, which are the probability values for the object landing on Surface 1 and Surface 2 respectively. Enter values between 0 and 1, make sure that the sum of P_1 and P_2 is equal to 1.

c. The recommended values for h_1 and h_2 will appear in the command window and the prompt will ask “Is the value okay?”. If they are type “ ‘y’ ” and if they are not type “ ‘n’ ”.

    i. If the vales are acceptable (‘y’), then the program will plot the desired percentage distribution on a histogram and a pop-up window will give the final values of h_1 and h_2.

    ii. If the values are not acceptable (‘n’), then the program will ask for acceptable values of h_1 and h_2 and then it will plot the percentage distribution according to the h_1 and h_2 values that are acceptable. A pop-up window will give the final values of h_1 and h_2.

d. The prompt will ask again if a change in the distribution is still desired. Repeat step 7.a.



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>> Start
Enter number of different orientations the object can have :
2
Enter lists of surface areas (mm^2) :
[2 4]
Enter lists of number of surfaces identical to and inclusive of the contacting surface :
[1 1]
Enter lists of distances from the surface to the center of gravity (mm) :
[3 5]

p =

   45.4545   54.5455

Change distribution ? :
   'y'
Enter the first percentage distribution 0<=p1<=1 :
   0.6
Enter the second percentage distribution 0<=p2<=1:
   0.4

 Recomended values:

 h1 = 2.000 , h2 = 6.000

 The desired distribution can be achieved by modifying the model so that
 the distances from the center of gravity correspond to the suggested h1
 and h2 values or any values that have the same h1 to h2 ratio


Is the value ok? :
 'n'
Enter desired distance from the surface to the center of gravity (h1) [mm] :
   1.5
Enter desired distance from the surface to the center of gravity (h2) [mm] :
   6.5

p =

   68.4211   31.5789

Change distribution ? :
   'n'
```

Fig. 85. Example of MATLAB interface for changing the distribution

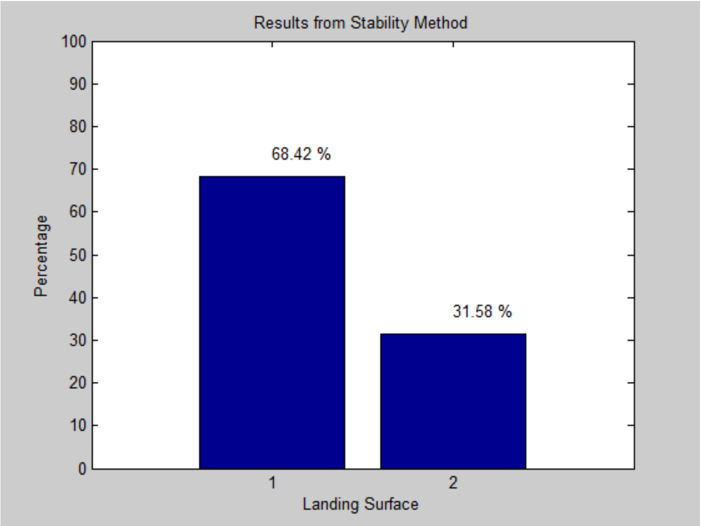
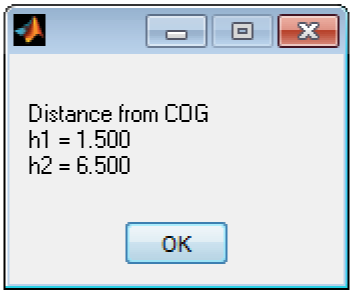


Fig. 86. Example of histogram obtained from change in distribution



Distance from COG
h1 = 1.500
h2 = 6.500

Fig. 87. Example of pop-up window showing values of h_1 and h_2

General rules of determining which values to consider

- **No extrusions** or **protrusions** such as prisms, where the number of surfaces on the object is equal to the number of stable orientations
  - Surface Area: Contacting surface area
  - Distance from COG: Distance from the center of the surface area to the COG
- **Curved surfaces**
  - Surface Area: consider the whole curved surface, not just the area that is in contact with the ground. Example shown in Fig. 88.

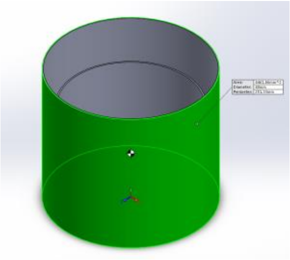


Fig. 88. Surface Area considered for curved surfaces high-lighted in green

  - Distance from COG: Minimum distance from the curved surface to the COG
- Extrusions or protrusions creating holes or uneven surfaces
  - Surface Area: Surface area of whole plane created from all the surfaces/ points that are directly in contact with the ground. An example is shown in Fig. 89.
  - Distance from COG: minimum distance from the center of the plane to the COG

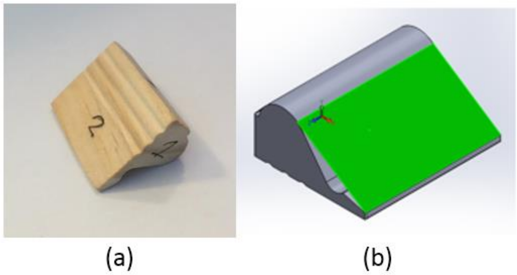


Fig. 89. Example of surface area consideration for objects with extrusions or protrusions creating holes or uneven surfaces; a) Stable orientation of object; b) Surface area considered for the stable orientation highlighted in green.

- Objects where number of surfaces is not equal to number of stable orientations
    - Surface Area: An unstable surface is a surface that the object can land on but cannot stabilize on. Add the surface area of the unstable surface to the surface area of the landing surface that the object stabilizes at after it hits the ground at the unstable landing surface.
        - Unstable surfaces of objects can come in two forms:
            - Plain flat surface (illustrated in Fig. 90)
            - Less obvious landing surfaces created by the complex geometry of the shape in question.
    - Distance from COG: Minimum distance from the center of the stable landing orientation

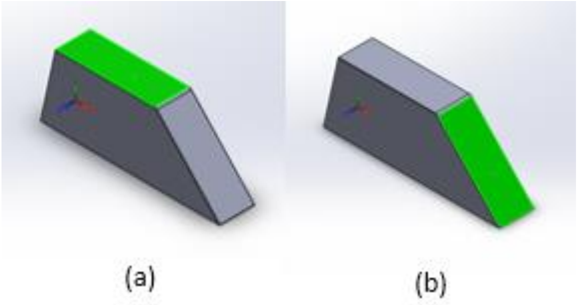


Fig. 90. Example of surface area considerations for objects with unstable landing surfaces, surfaces highlighted in green; a) Stable surface 1; b) unstable surface that will stabilize at stable surface 1

Example: Simplified model of the product of interest

The product has two different stable landing surfaces.

Surface 1

- The surface area considered for the stable landing surface, Surface 1 is shown in Fig. 91. A plane was created to account for both the area that is directly in contact with the ground and the area of the two extrusions in the center of the model. The whole plane is considered as part of the value for the surface area.

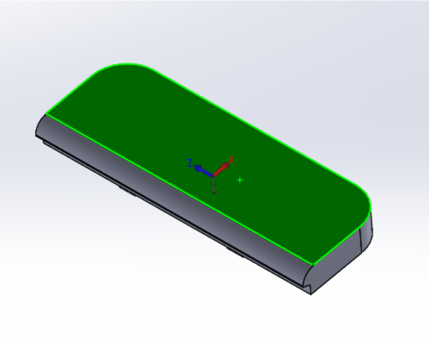


Fig. 91. Surface Area considered for Surface 1 of the simplified product

The distance of the surface from the COG used for this is the distance from the COG from the center of the plane.

Surface 2

- Five different landing surface areas were considered for Landing Surface 2.
- The main surface that the object stabilizes on is shown in Fig. 92. The value for the distance from the center of gravity is taken from for Landing Surface 2 (the surface distance from the center of gravity is always taken from the center of the stable landing surface).

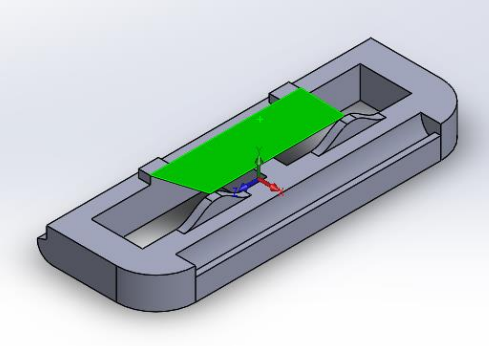


Fig. 92. Surface Area of stable landing surface considered for Surface 2 of the simplified product

- The second surface area considered for Landing Surface 2 is the unstable landing surface 1, shown in Fig. 93. The points selected to create the plane shown in Fig. 93. (a) is shown in Fig. 93. (b).

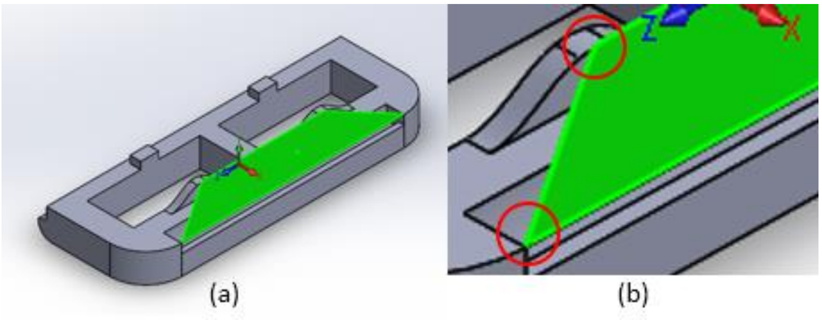


Fig. 93. Surface Area of unstable landing surface 1 considered for Surface 2 of the simplified product; (a) Overview of landing surface; (b) Detailed view showing points to select.

- Since the product of interest is symmetrical against the x-axis therefore the plane for the landing surface can be constructed on one side and multiplied by two. For the purpose of this case study, both sides were shown. The unstable landing surfaces that this applies to are unstable landing surfaces 2, 3 and 4.
  - The unstable landing surfaces 2, 3 and 4 are shown in Fig. 94, Fig. 95, and Fig. 96 with (a) showing the over view of the surface and (b) showing the points to select to create the plane for the landing surfaces shown.

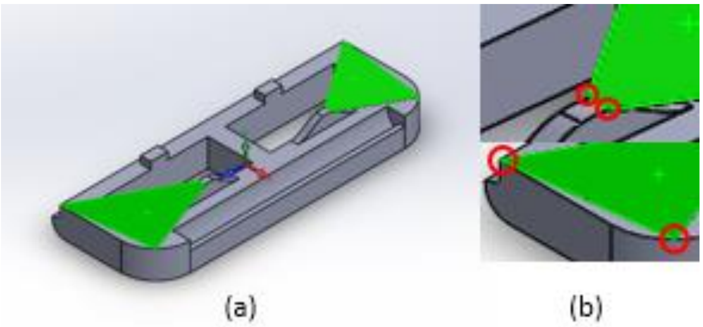


Fig. 94. Surface Area of unstable landing surface 2 considered for Surface 2 of the simplified product; (a) Overview of landing surface; (b) Detailed view showing points to select.

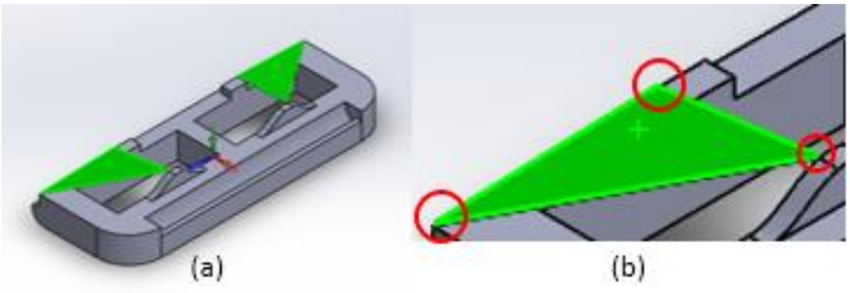


Fig. 95. Surface area of unstable landing surface 3 considered for Surface 2 of the simplified product; (a) Overview of landing surface; (b) Detailed view showing points to select.

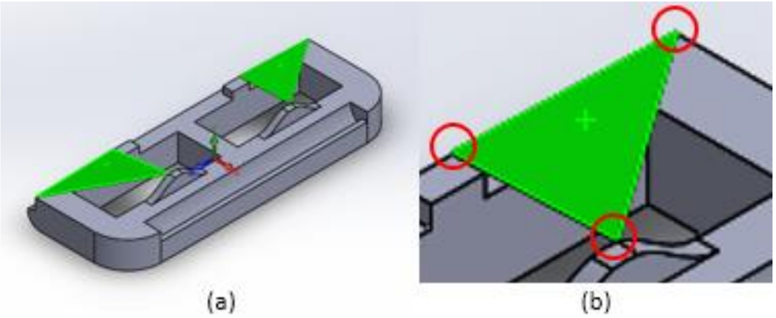


Fig. 96. Surface Area of unstable landing surface 4 considered for Surface 2 of the simplified product; (a) Overview of landing surface; (b) Detailed view showing points to select.

- The fifth surface area considered is shown in Fig. 97. The points needed to create the plane are shown in Fig. 97. (b), select the same points of the respective sides.

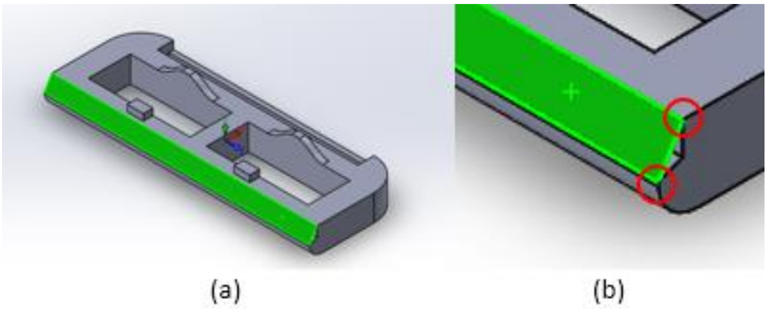


Fig. 97. Surface are of unstable landing surface 5 considered for Surface 2 of the simplified product; (a) Overview of landing surface; (b) Detailed view showing points to select.

The surface area and the percentage occurrences obtained from the simplified product using the Stability method can be found in Table 14 in Appendix B and Fig. 24 in Section 3.2.5.

Appendix K. Computational Method Python Script

The script is not included in this report. A soft copy of the script can be obtained from either Professor Cosme Furlong at WPI or Loren Gjata.

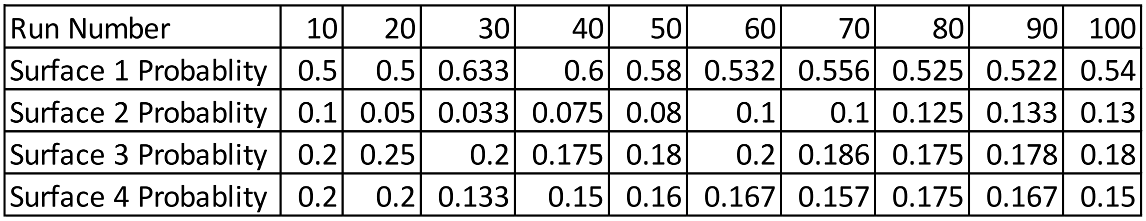
Appendix L. Computational Method MATLAB Script

The script is not included in this report. A soft copy of the script can be obtained from either Professor Cosme Furlong at WPI or Loren Gjata.

Appendix M. Convergence Test Results

As with any probability data the accuracy of the final results is highly dependent on the sample size chosen. In order to ensure we had a large enough sample size yet not too many samples were taken, due to the computation cost of running these tests. An initial "convergence" test was done. This was done by recording the probabilities of the surfaces by 10 drop increments. The results of this test show when the results start to converge to the true value. These results can be seen in Table 25 and Fig. 98.

Table 25: Results of landing surface probability over 100 runs



Run Number	10	20	30	40	50	60	70	80	90	100
Surface 1 Probablity	0.5	0.5	0.633	0.6	0.58	0.532	0.556	0.525	0.522	0.54
Surface 2 Probablity	0.1	0.05	0.033	0.075	0.08	0.1	0.1	0.125	0.133	0.13
Surface 3 Probablity	0.2	0.25	0.2	0.175	0.18	0.2	0.186	0.175	0.178	0.18
Surface 4 Probablity	0.2	0.2	0.133	0.15	0.16	0.167	0.157	0.175	0.167	0.15

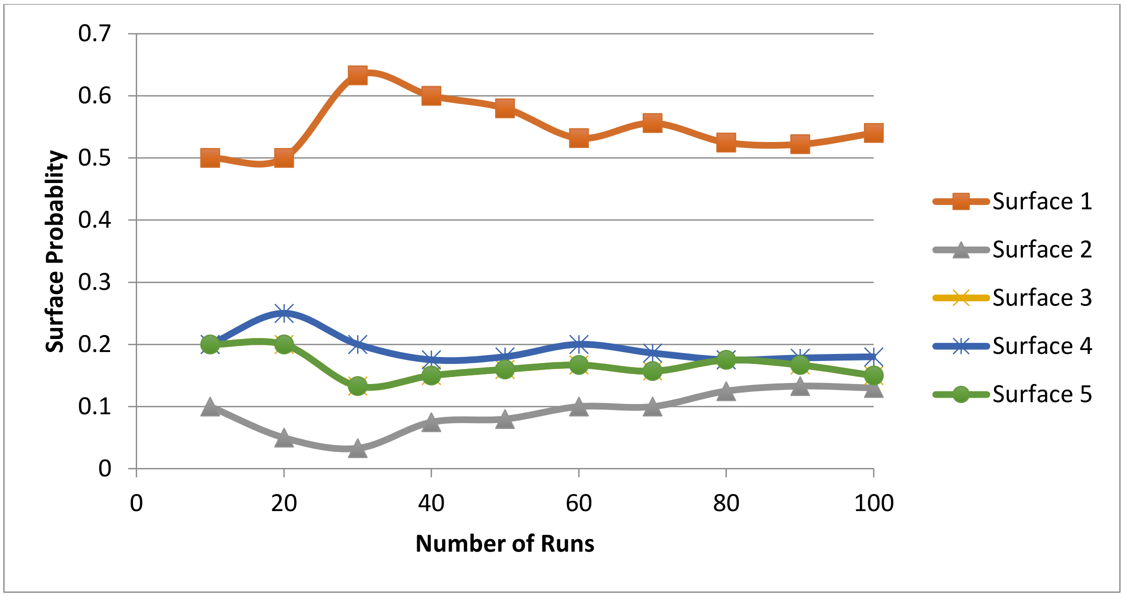


Fig. 98. Graphical results of landing surface probability over 100 runs

Another way to observe these results are to graph there average deviance from the physical testing results. This is shown in Fig. 99.

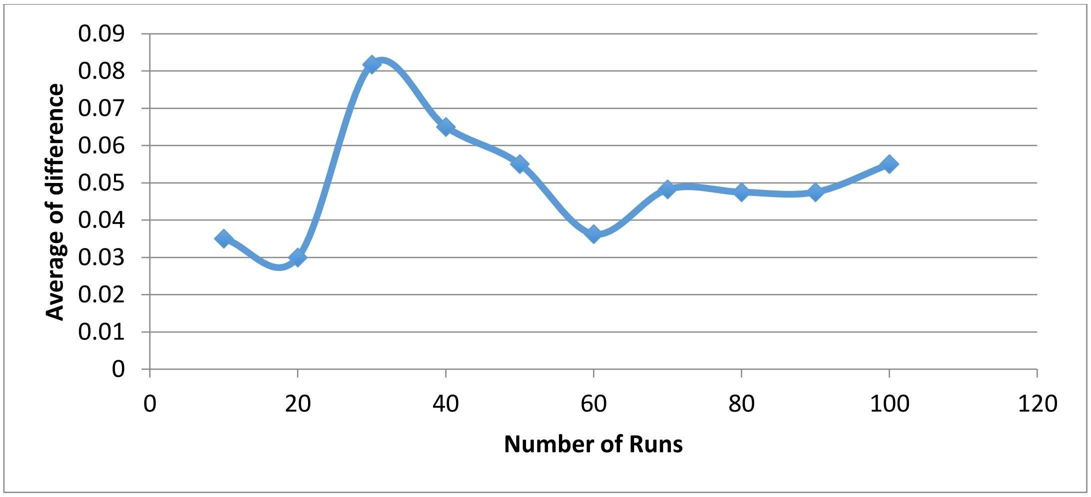


Fig. 99. Graphical representation of average difference with respects to the physical results over 100 runs

These figures show that once the run number hits around 70 runs the probabilities start to converge to the true value of the simulation. With this being said it can be conclude that for our purposes, given time and resource constraints, that running 70 runs will provide us the best results for the time taken.

Appendix N. User Manual for Meshing

Prerequisites

Before starting to mesh the part, the user needs to make sure that **the following requirements have been met**:

(a). A computer with ABAQUS Version 6.14-1 or higher.

(b). A step file of selected part modeled in meters.

Instructions

1. Start ABAQUS Software and perform preliminary setup

    a.) Double Click ABAQUS CAE icon, as shown in Fig. 100, to start ABAQUS software

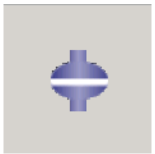


Fig. 100. ABAQUS CAE

    b.) After ABAQUS has been launched, selecting "With Standard/Explicit Model" in the Start Session Window, as shown in Fig. 101.

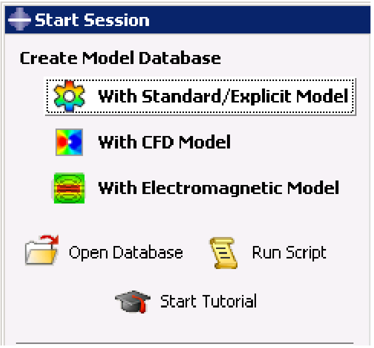


Fig. 101. Select "With Standard/Explicit Model"

2. Set the working directory

a) Go to 'File' > 'Set Work Directory', as shown in Fig. 102.

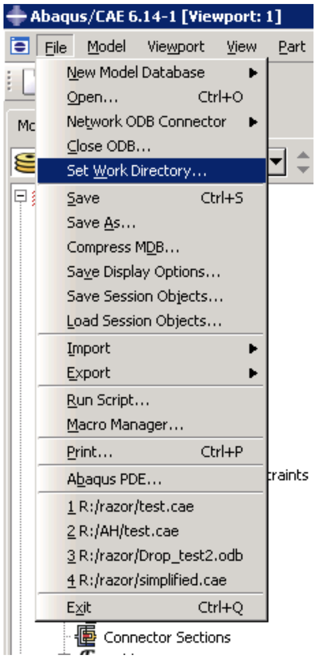
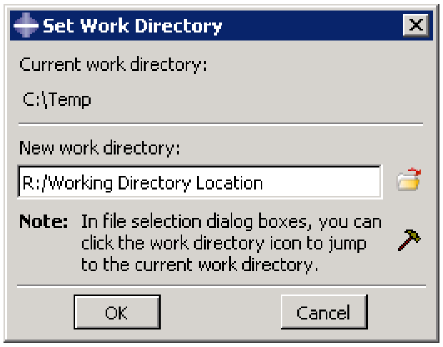


Fig. 102.Setting working directory in ABAQUS

b) Select a destination where all the work will be saved, and click Ok, as shown in Fig. 103

This should be the place the dropping model step file is located.



Set Work Directory

Current work directory:

C:\Temp

New work directory:

R:/Working Directory Location

Note: In file selection dialog boxes, you can click the work directory icon to jump to the current work directory.

Fig. 103. Selection of working directory

3. Import model into ABAQUS. (‘File’ > ‘Import’ > ‘Part’) As shown in Fig. 104.

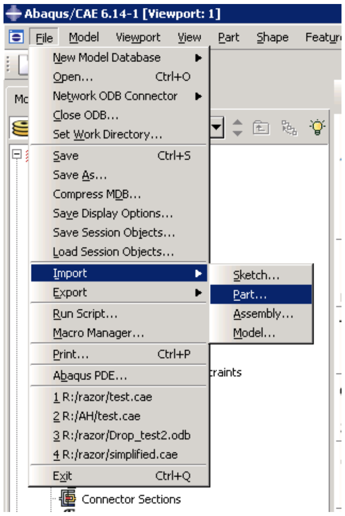
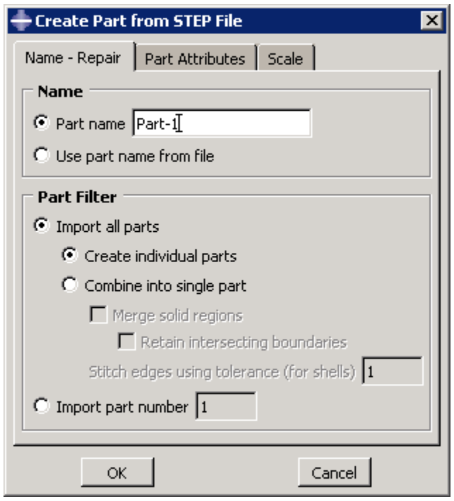


Fig. 104. How to import parts from the menu

4. Once prompted by dialog shown in Fig. 105, change part name to Part-1, and hit ok.



Create Part from STEP File

Name - Repair | Part Attributes | Scale

Name

● Part name: Part-1

○ Use part name from file

Part Filter

● Import all parts

● Create individual parts

○ Combine into single part

□ Merge solid regions

□ Retain intersecting boundaries

Stitch edges using tolerance (for shells): 1

○ Import part number: 1

OK | Cancel

Fig. 105. Part creation dialog

5. Expand model and double click Mesh (Empty) as shown in Fig. 106.

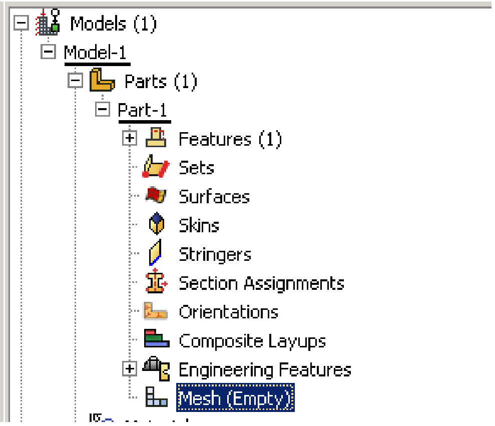


Fig. 106. Mesh location in the models files.

6. Select ‘Element Type’ under the ‘Mesh’ tab.

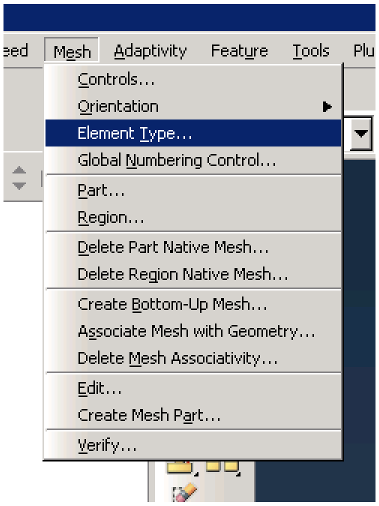
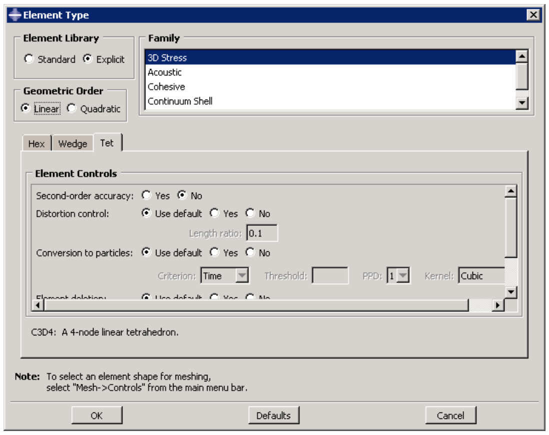


Fig. 107. Element Type location

7. Select an Explicit Standard element type; this should correspond to an element of C3D4. This is shown in Fig. 108.



Element Type

Element Library: ○ Standard ● Explicit

Geometric Order: ● Linear ○ Quadratic

Family: 3D Stress (selected), Acoustic, Cohesive, Continuum Shell

Hex | Wedge | Tet

Element Controls

Second-order accuracy: ○ Yes ● No

Distortion control: ● Use default ○ Yes ○ No

Length ratio: 0.1

Conversion to particles: ● Use default ○ Yes ○ No

Criterion: Time Threshold: PPD: 1 Kernel: Cubic

Element deletion: ● Use default ○ Yes ○ No

C3D4: A 4-node linear tetrahedron.

Note: To select an element shape for meshing, select "Mesh->Controls" from the main menu bar.

Fig. 108. Element Type selection of a C3D4 structure

8. Select ‘Mesh’ > ‘Controls’, shown in Fig. 109.

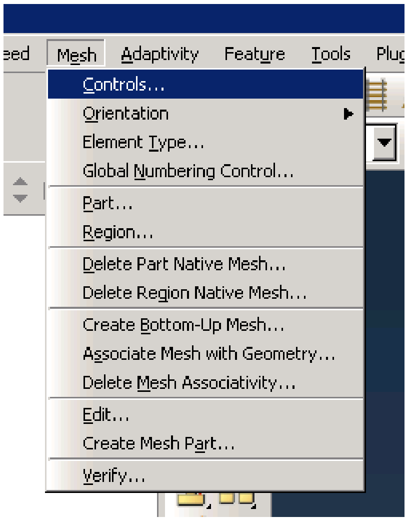
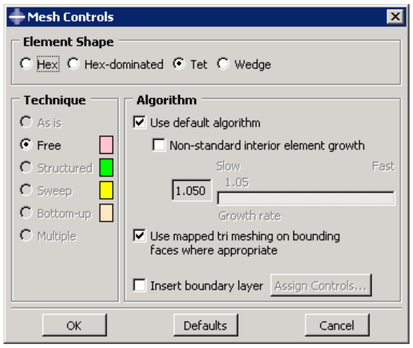


Fig. 109. Location of mesh controls

9. Once you select mesh controls click on part and hit enter, then a prompt shown in Fig. 110 will appear. Select ‘Tet’ for element shape and hit enter.



Mesh Controls

Element Shape

○ Hex ○ Hex-dominated ● Tet ○ Wedge

Technique

○ As is
● Free 🟥
○ Structured 🟩
○ Sweep 🟨
○ Bottom-up □
○ Multiple

Algorithm

☑ Use default algorithm

□ Non-standard interior element growth

Slow Fast 1.050 1.05 Growth rate

☑ Use mapped tri meshing on bounding faces where appropriate

□ Insert boundary layer

Fig. 110. Selection of element shape

10. Select ‘Seed’ > ‘Part’, when prompted use default options and hit ok, as shown in Fig.111.

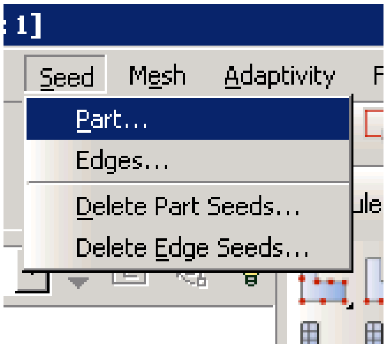


Fig. 111. Selection of seed size

11. Mesh part using ‘Mesh’> ‘Part’, then hit enter as shown in Fig. 112.

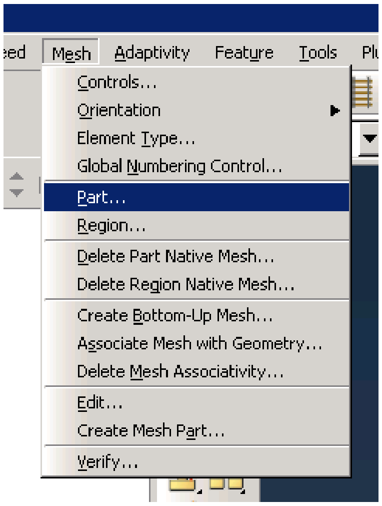


Fig. 112. Location of mesh part.

12. Double click surfaces as shown in Fig. 113.

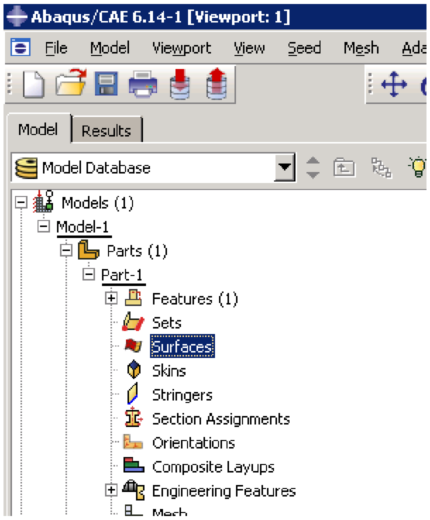


Fig. 113. Location of Surfaces in model

13. When prompted by surface name hit continue.

14. Select possible landing face and hit enter.

15. Repeat steps 12-14 until all the landing surfaces are accounted for. Examples of important vs non-important surfaces are shown in Fig.114 and Fig. 115.

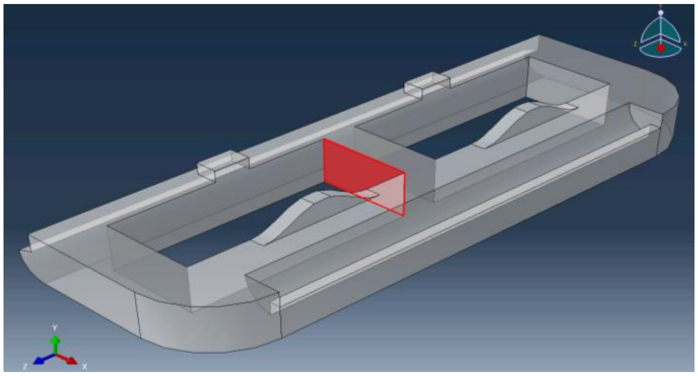


Fig. 114 Surface in red is not an important surface to label since it will never come into contact with the ground.

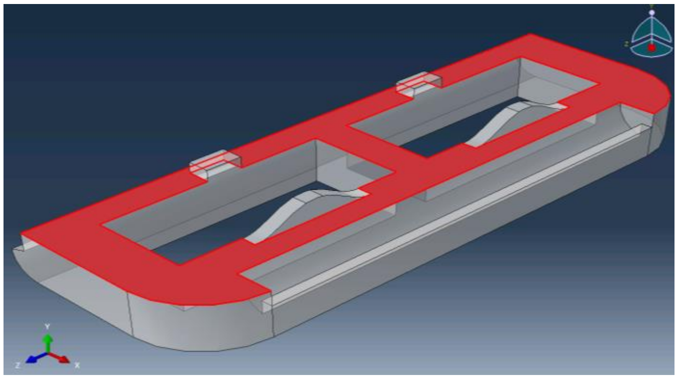
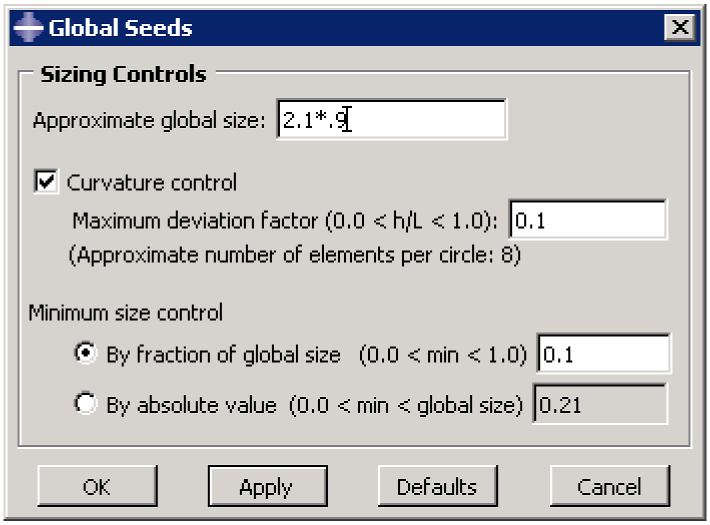


Fig. 115 Surface in red is an important surface as it has a chance to come into contact with the ground.

16. Save file.

17. Run saved .CAE through script for one run to get landing surface.

18. Repeat steps 10-11 on same file, this time change default seed size by .90, as shown in Fig. 116



Global Seeds

Sizing Controls

Approximate global size: 2.1*.9

☑ Curvature control

Maximum deviation factor (0.0 < h/L < 1.0): 0.1

(Approximate number of elements per circle: 8)

Minimum size control

● By fraction of global size (0.0 < min < 1.0) 0.1

○ By absolute value (0.0 < min < global size) 0.21

Fig. 116. Changing of seed size in parts mesh

19. Save file and run through script again for 1 run.

20. Repeat step 18, 19 but with seed size modifier of .85, .80, .70, and .65.

21. Find the largest seed size modifier where the parts final orientation matches the orientation obtained from .65.

22. Repeat steps 10-11 with the selected seed size.

23. Save file

24. The file is now ready for full analysis using the script.

Appendix O. User Manual for Computational Tool

Prerequisites

Before starting running either of the scripts, the user needs to make sure that **the following requirements have been met**:

(a). A computer with ABAQUS Version 6.14-1 or higher and MATLAB 8.1 (R2013a) or higher installed.

(b). A .CAE file which contains the following:

i. The dropping model has been accurately meshed. A detailed instruction can be found in the Appendix N.

ii. The contact surfaces of dropping model are correctly defined and labeled. A detailed instruction can be found in Appendix N. Record the number of labeled contact surfaces, which will be used as a user input during the analysis.

iii. The material properties (Young's modulus [Pa], Poisson's ratio and Density [Kg/m^3]) for the dropping model and ground model.

- If the user decides to assign multiple materials to the dropping model, this must be done manually to define sections and assign corresponding materials.
- If the user decides to assign a single material to the dropping model, this can be done via user inputs during running the automation tools.

(d). The path names to the accurately meshed dropping model (.CAE file) and ground model (.STEP file).

i. To copy the full path for an individual file, hold down the Shift key as you right click the file, and then choose Copy As Path.

ii. These path names can be saved into a text file, and then copies and pasted into user inputs dialog boxes during running the automation tools.

(e). Three Python Scripts: (i). Main. Py; (ii). ReadText.py are available in ABAQUS working directory.

(f). Ground model step file titled with "Ground.STEP" is available in Abaqus working directory.

(g). Two MATLAB Scripts: (i). Run.m; (ii) SURResult.m are available in MATLAB working directory.

If all the requirements are met, the user can go to the next page. Follow the instructions and start running the program.

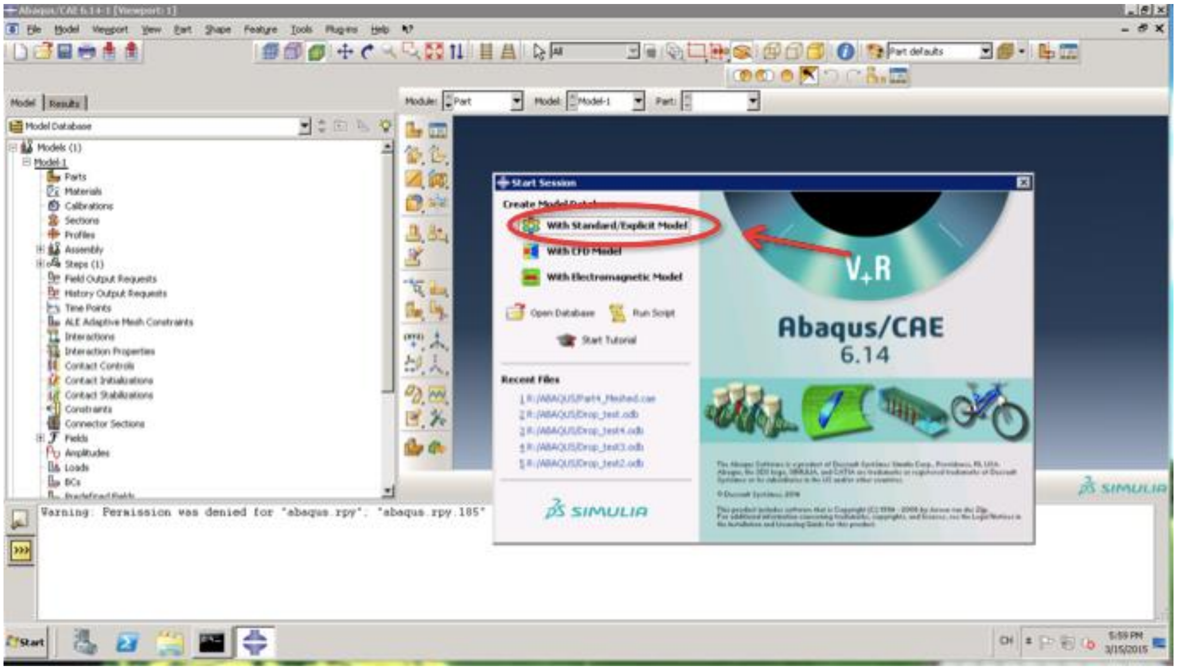
Instructions

1. Start ABAQUS Software and perform preliminary setup

   a) Double Click ABAQUS CAE icon, as shown in Fig. 117, to start ABAQUS software.

Fig. 117. ABAQUS CAE

   b) After ABAQUS has been launched, select ‘With Standard/Explicit Model’ in the Start Session Window, as shown in Fig. 118.

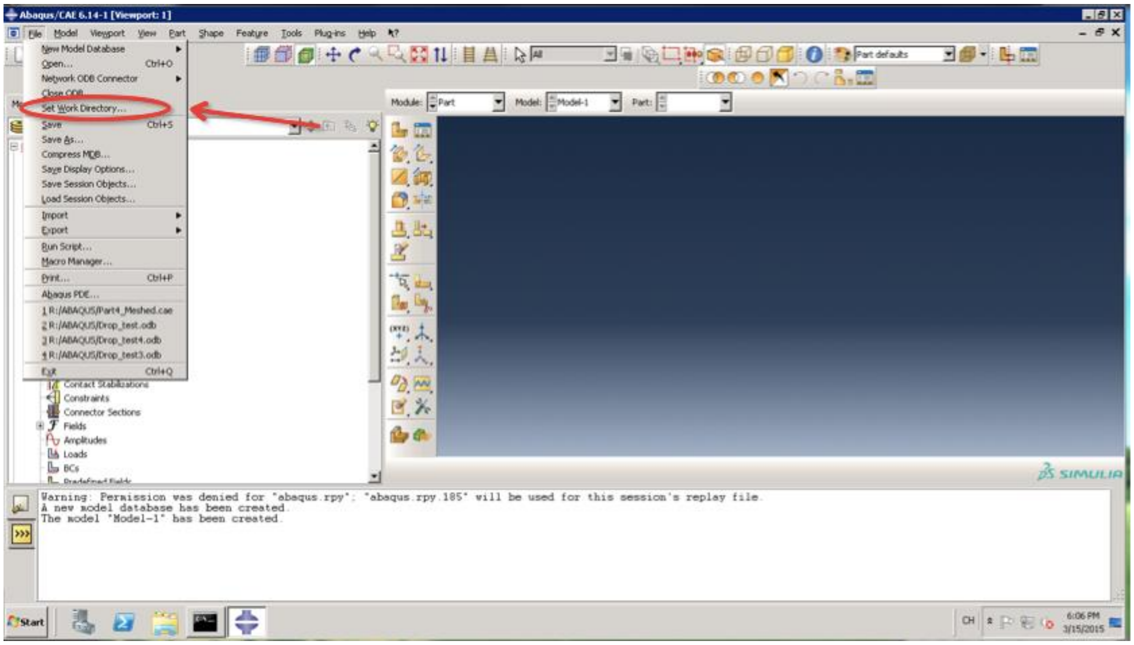


```
Warning: Permission was denied for "abaqus.rpy"; "abaqus.rpy.185"
```

Fig. 118. Select "With Standard/Explicit Model"

2. Set the working directory

c) Go to ‘File’ > ‘Set Work Directory’, as shown in Fig.119.

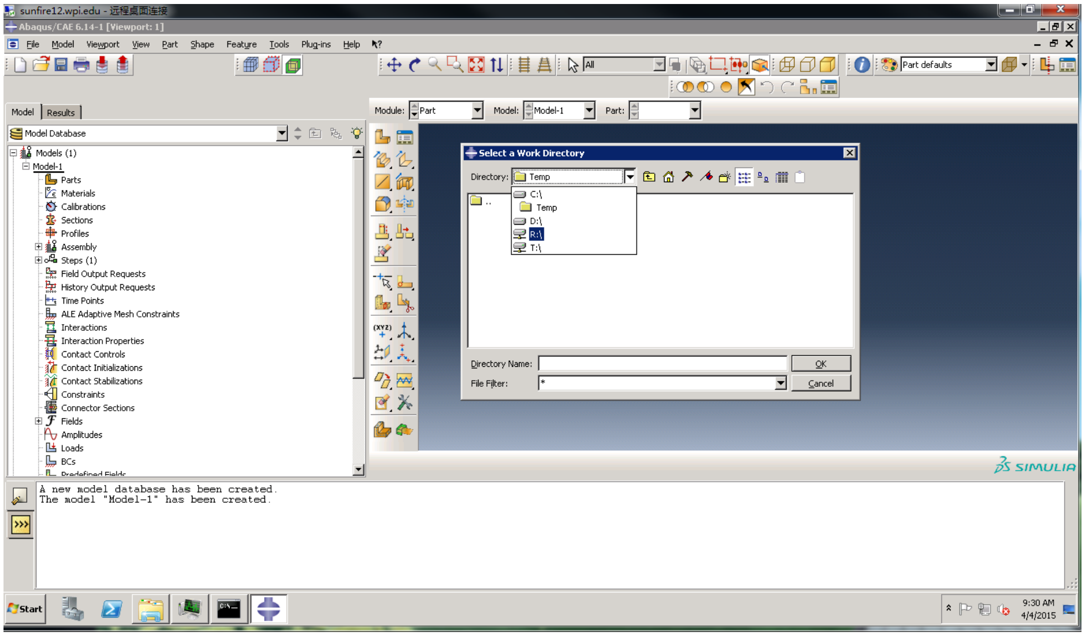


```
Warning: Permission was denied for "abaqus.rpy"; "abaqus.rpy.185" will be used for this session's replay file.
A new model database has been created.
The model "Model-1" has been created.
```

Fig. 119. Set Work Directory

d) Select a destination where all the work will be saved, and click Ok, as shown in Fig. 120.

This should be the place where contains all Python Scripts and Ground model step file.



Module: Part Model: Model-1 Part:

Directory: Temp

Directory Name:

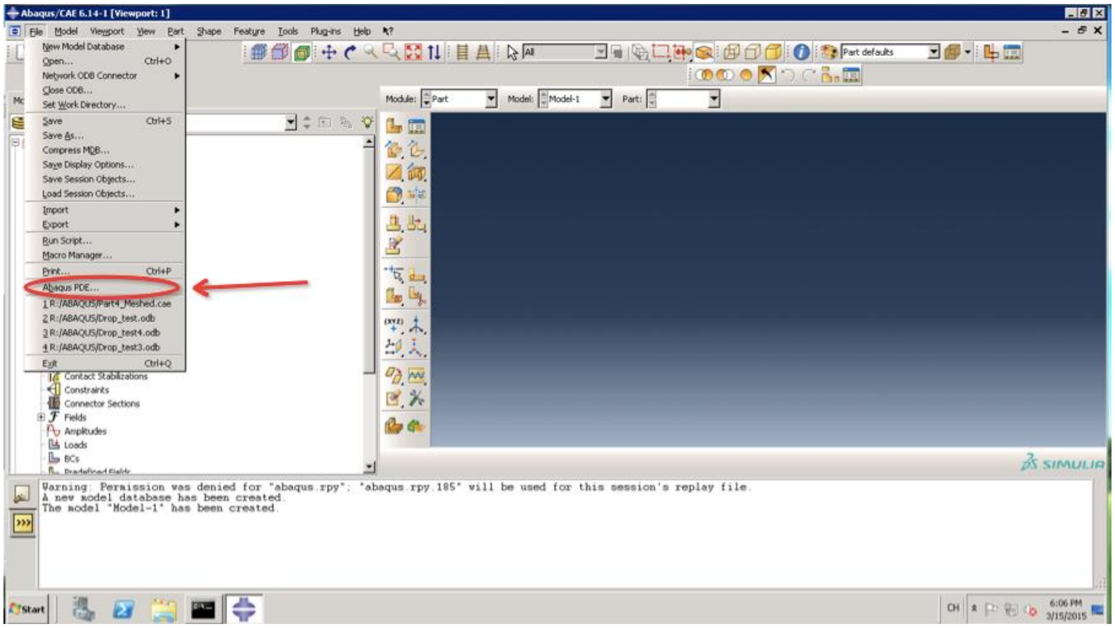
File Filter: *

```
A new model database has been created.
The model "Model-1" has been created.
```

Fig. 120. Set Work Directory

3. Launch ABAQUS PDE

a) Go to ‘File’ > ‘ABAQUS PDE’, as shown in Fig.121.

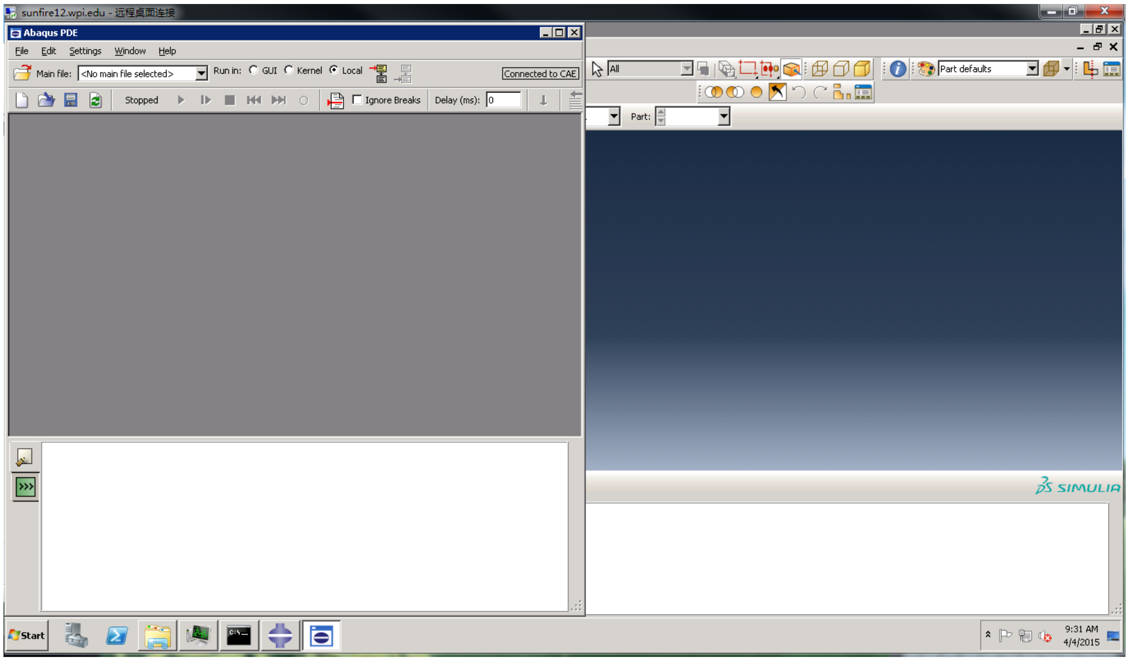


Module: Part Model: Model-1 Part:

```
Warning: Permission was denied for "abaqus.rpy"; "abaqus.rpy.185" will be used for this session's replay file.
A new model database has been created.
The model "Model-1" has been created.
```

Fig. 121. ABAQUS PDE\

b) Successful Launching of ABAQUS PDE is shown in Fig.122.


sunfire12.wpi.edu - 远程桌面连接
Abaqus PDE
File Edit Settings Window Help
Part defaults
SIMULIA
Start

Main file: <No main file selected> Run in: ○ GUI ○ Kernel ● Local Connected to CAE

Stopped □ Ignore Breaks Delay (ms): 0

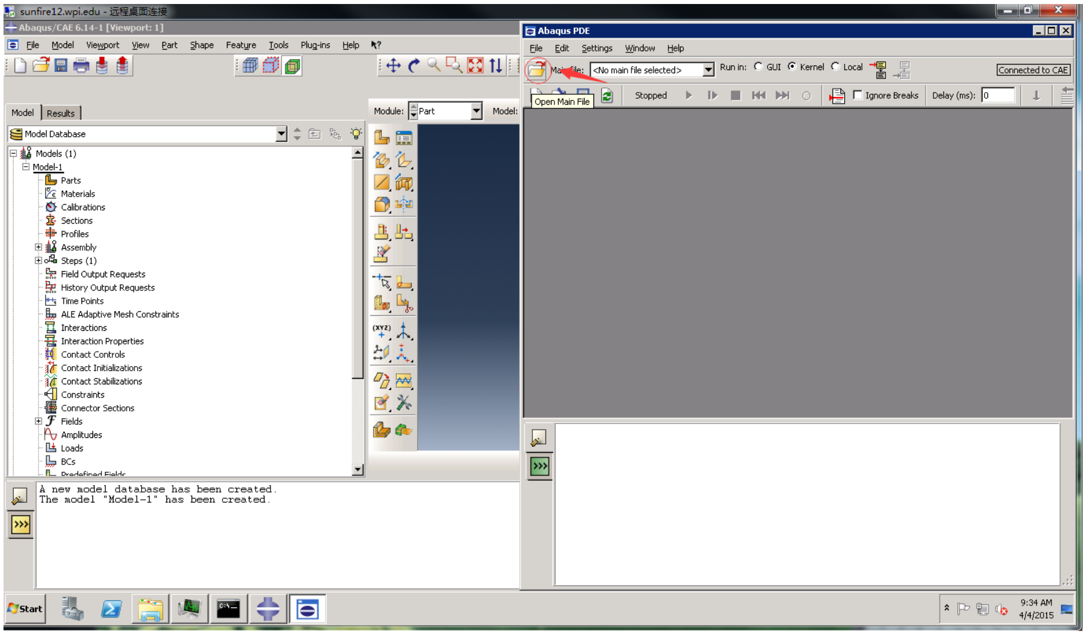
Part:

Fig. 122. Launch ABAQUS PDE successfully

4. Open Python Script

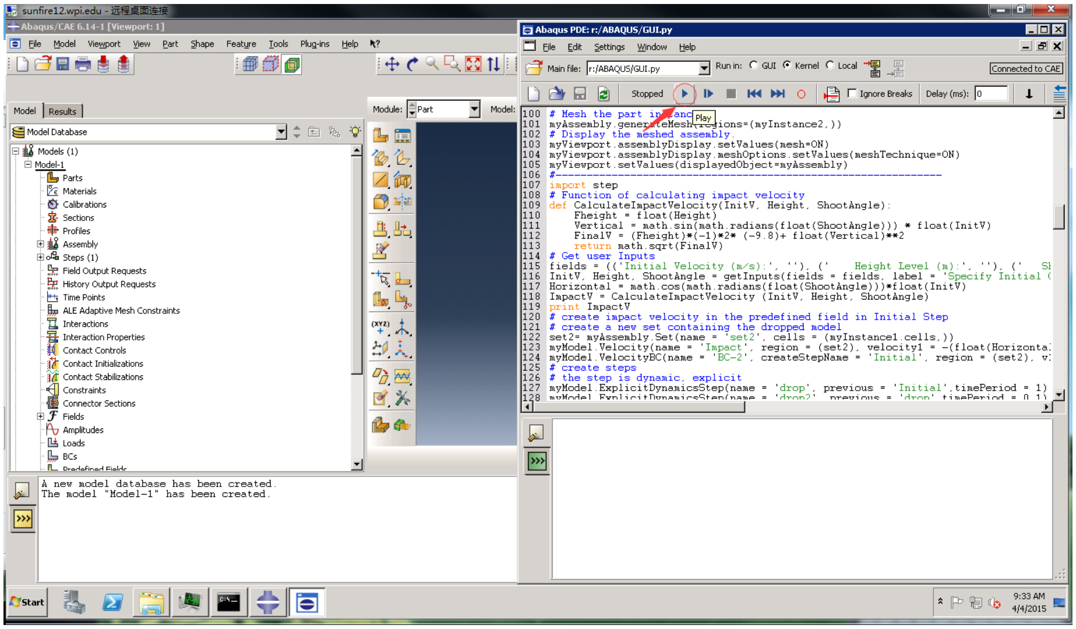
a) Go to Open Main File as shown in Fig.123, then select the Python script titled “Main.py”.

b) The selected Python script will be shown in the window. Click Play to start running the script as shown in Fig. 124.



```
A new model database has been created.
The model "Model-1" has been created.
```

Fig. 123. Open Main File



Main file: r:/ABAQUS/GUI.py

Run in: ○ GUI ● Kernel ○ Local

☐ Ignore Breaks Delay (ms): 0

```
# Mesh the part instance
myAssembly.generateMesh(regions=(myInstance2,))
# Display the meshed assembly.
myViewport.assemblyDisplay.setValues(mesh=ON)
myViewport.assemblyDisplay.meshOptions.setValues(meshTechnique=ON)
myViewport.setValues(displayedObject=myAssembly)
#---------------------------------------------------------------
import step
# Function of calculating impact velocity
def CalculateImpactVelocity(InitV, Height, ShootAngle):
    Fheight = float(Height)
    Vertical = math.sin(math.radians(float(ShootAngle))) * float(InitV)
    FinalV = (Fheight)*(-1)*2* (-9.8)+ float(Vertical)**2
    return math.sqrt(FinalV)
# Get user Inputs
fields = (('Initial Velocity (m/s):', ''), ('    Height Level (m):', ''), ('   Sh
InitV, Height, ShootAngle = getInputs(fields = fields, label = 'Specify Initial (
Horizontal = math.cos(math.radians(float(ShootAngle)))*float(InitV)
ImpactV = CalculateImpactVelocity (InitV, Height, ShootAngle)
print ImpactV
# create impact velocity in the predefined field in Initial Step
# create a new set containing the dropped model
set2= myAssembly.Set(name = 'set2', cells = (myInstance1.cells,))
myModel.Velocity(name = 'Impact', region = (set2), velocity1 = -(float(Horizonta
myModel.VelocityBC(name = 'BC-2', createStepName = 'Initial', region = (set2), v
# create steps
# the step is dynamic, explicit
myModel.ExplicitDynamicsStep(name = 'drop', previous = 'Initial',timePeriod = 1)
myModel.ExplicitDynamicsStep(name = 'drop2', previous = 'drop',timePeriod = 0.1)
```

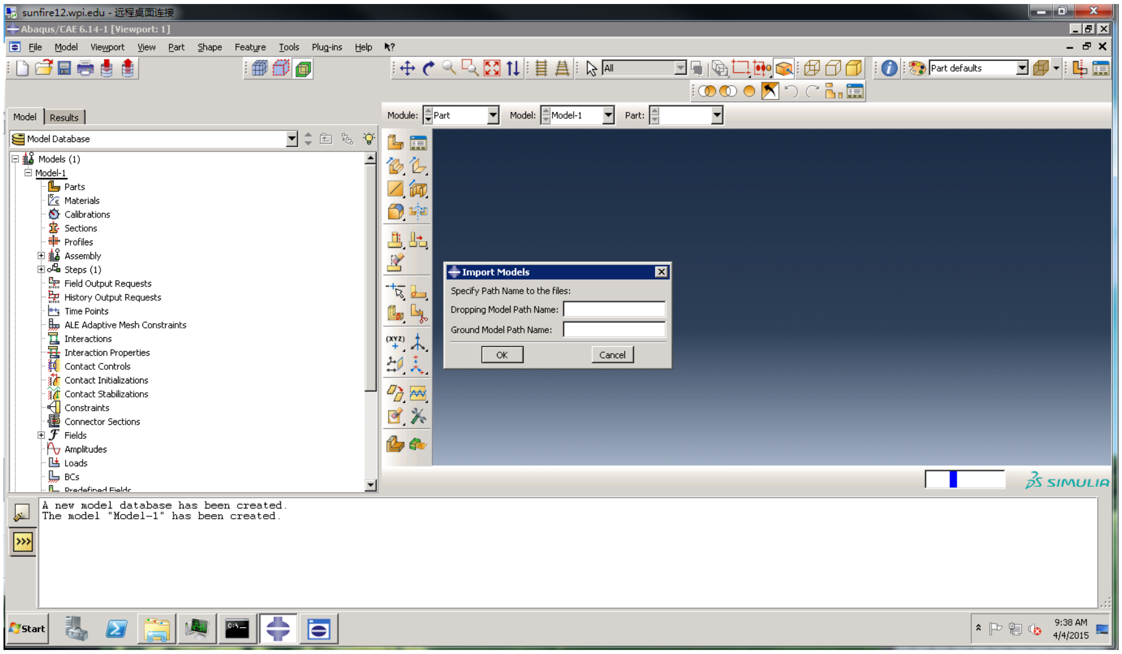
```
A new model database has been created.
The model "Model-1" has been created.
```

Fig. 124. Run Python Script

5. Enter user inputs

*Note: The user now can minimize the ABAQUS PDE window to see the user input dialog prompts.

a) Copy and paste the path name to the dropping model and ground model that are saved at the beginning in its corresponding box, as shown in Fig. 125.



Specify Path Name to the files:

Dropping Model Path Name:

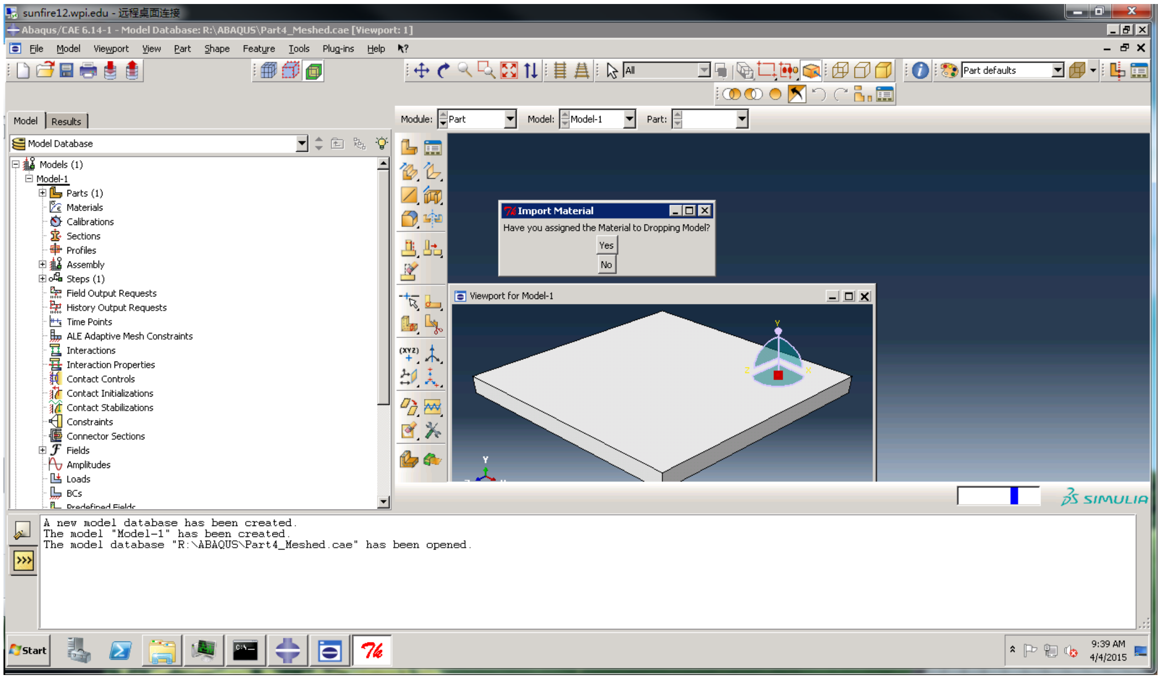
Ground Model Path Name:

```
A new model database has been created.
The model "Model-1" has been created.
```

Fig. 125. Import Models

b) The user needs to choose "Yes" or "No", as shown in Fig. 126.

   i. "Yes" – if the user has assigned materials to the dropping model, only the material to the ground model needs to be assigned as shown in Fig. 127. Please enter the Material name, Young's modulus, Poisson's ratio and density based on the information prepared at the beginning.

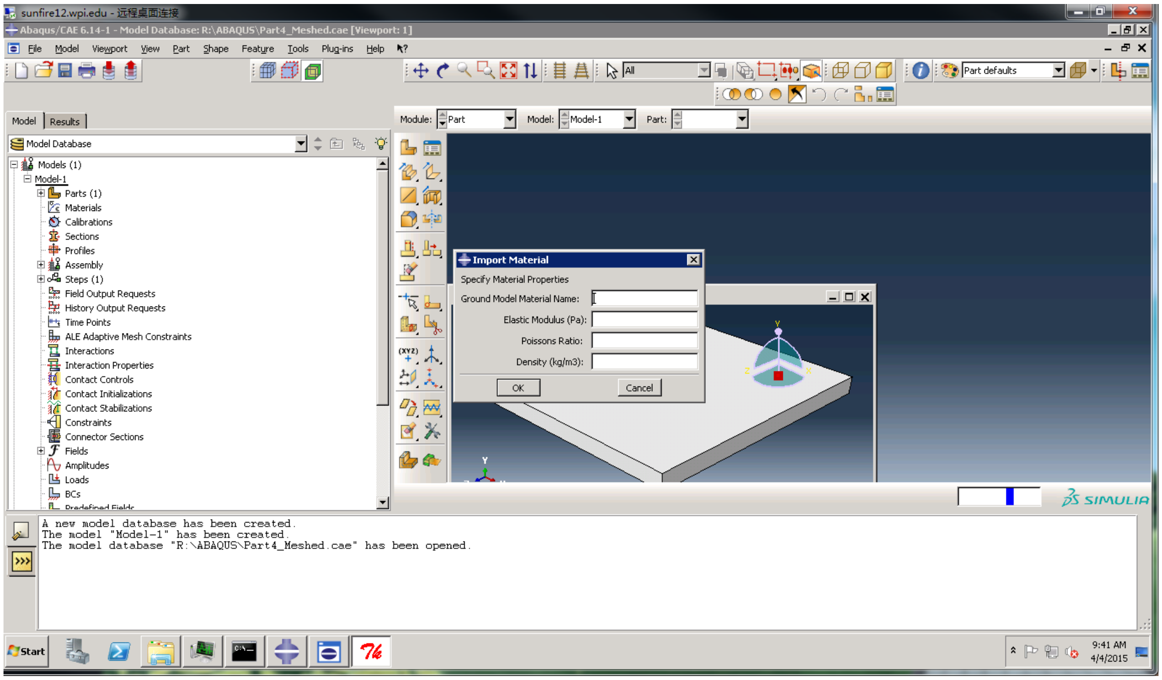


Import Material

Have you assigned the Material to Dropping Model?

```
A new model database has been created.
The model "Model-1" has been created.
The model database "R:\ABAQUS\Part4_Meshed.cae" has been opened.
```

Fig. 126. Assign Materials



Import Material

Specify Material Properties

Ground Model Material Name:

Elastic Modulus (Pa):

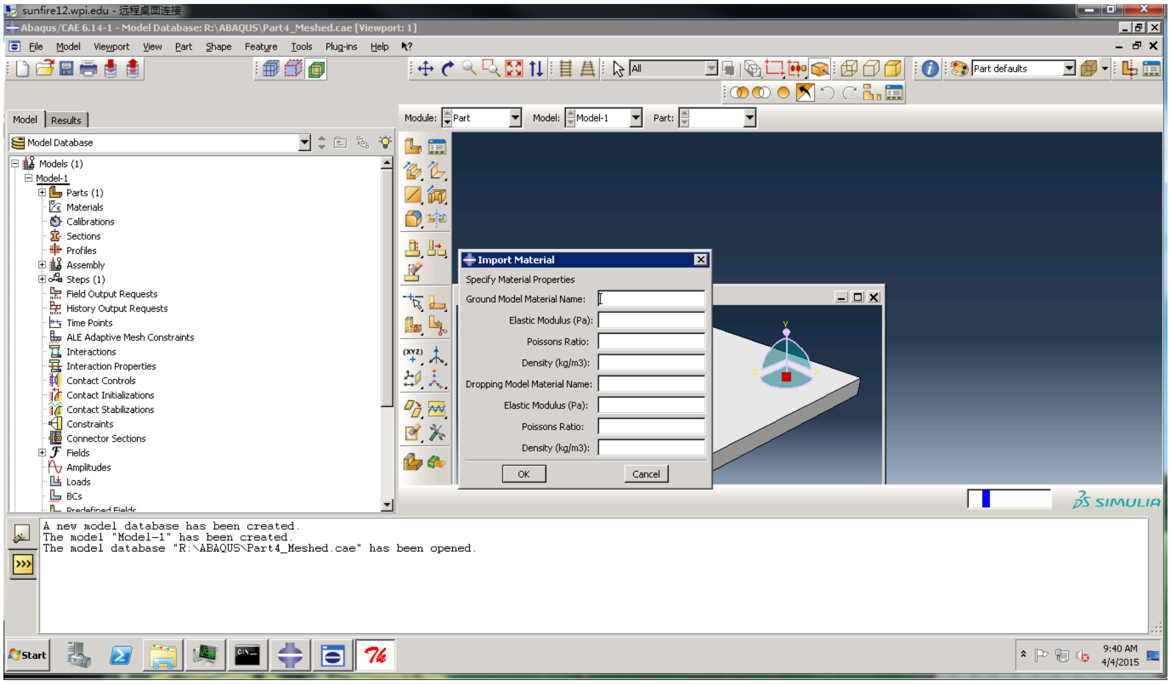
Poissons Ratio:

Density (kg/m3):

```
A new model database has been created.
The model "Model-1" has been created.
The model database "R:\ABAQUS\Part4_Meshed.cae" has been opened.
```

Fig. 127. Choose “Yes”

ii. "No" – if the user has not assigned materials to the dropping model, the materials to the dropping and the ground model needs to be assigned as shown in Fig. 128. Please enter the Material name, Young's modulus, Poisson's ratio and density based on the information prepared at the beginning.



Import Material

Specify Material Properties

Ground Model Material Name:

Elastic Modulus (Pa):

Poissons Ratio:

Density (kg/m3):

Dropping Model Material Name:

Elastic Modulus (Pa):

Poissons Ratio:

Density (kg/m3):

OK Cancel

```
A new model database has been created.
The model "Model-1" has been created.
The model database "R:\ABAQUS\Part4_Meshed.cae" has been opened.
```

Fig. 128. Choose "No"

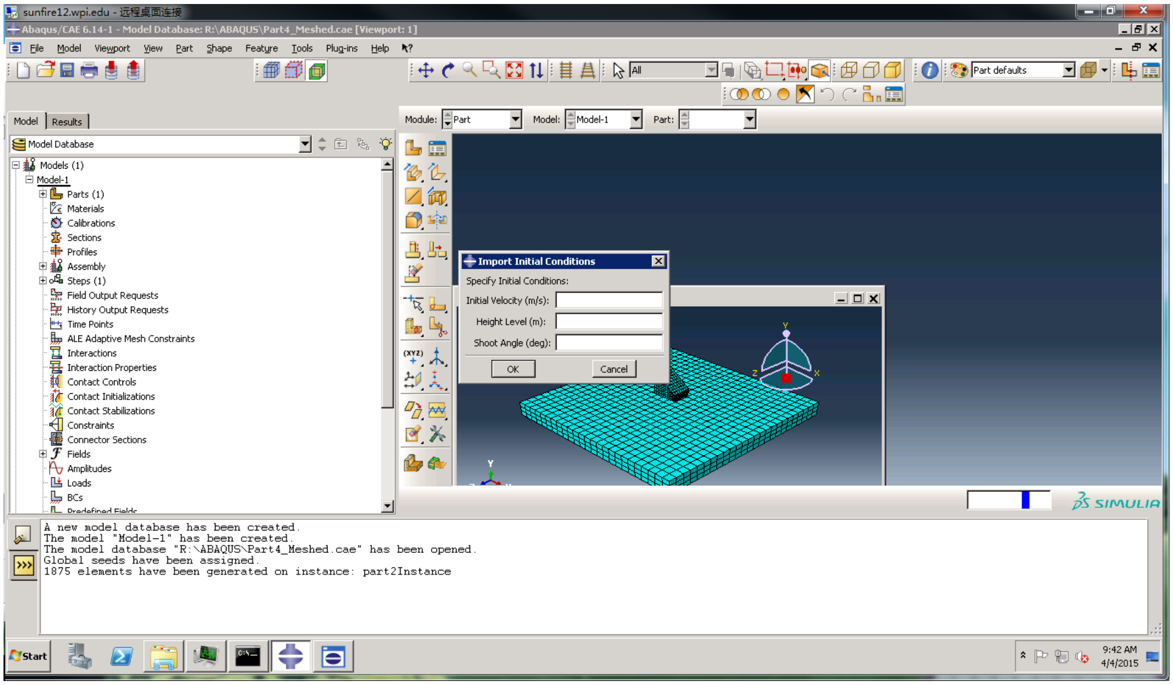
c) Import initial conditions

i. Please enter the values for initial dropping velocity, height and angle based on user's own choice, as shown in Fig.129. An example of inputs is shown as following:

- Initial Velocity (m/s): - 0.1
- Initial Height (m): 0.2

- Initial Shoot Angle (Degree): 85

*Note: the value of initial velocity should be a negative number, and initial height and shoot angle should be positive numbers. The range of inputs for initial shoot angle should be between 60 and 90 degrees.



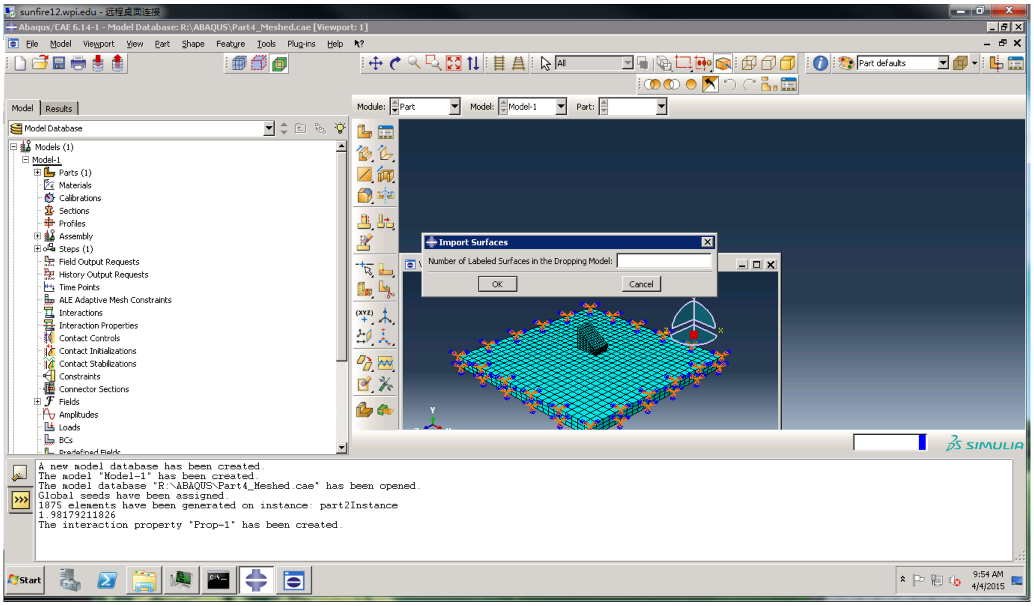
```
A new model database has been created.
The model "Model-1" has been created.
The model database "R:\ABAQUS\Part4_Meshed.cae" has been opened.
Global seeds have been assigned.
1875 elements have been generated on instance: part2Instance
```

Fig. 129. Import Initial Conditions

d) Import number of labeled surfaces.

i. Please enter the information prepared at the beginning as shown in Fig. 130. The value should be the number of labeled contact surfaces in the dropping model.

Until this step, the first ABAQUS input file is created. The following procedures are involved in modifying the input file and generate series of new input files to submit for analysis.



Module: Part
Model: Model-1
Part:

Import Surfaces

Number of Labeled Surfaces in the Dropping Model:

OK

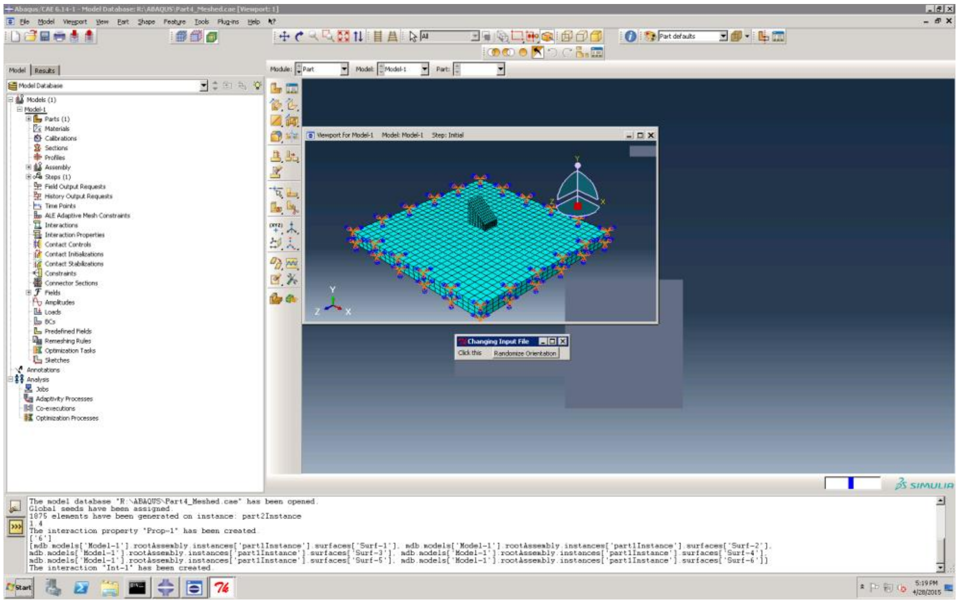
Cancel

```
A new model database has been created.
The model "Model-1" has been created.
The model database "R:\ABAQUS\Part4_Meshed.cae" has been opened.
Global seeds have been assigned.
1875 elements have been generated on instance: part2Instance
1.98179211826
The interaction property "Prop-1" has been created.
```

Fig. 130. Import Number of Labeled Surfaces

6. Modify input file

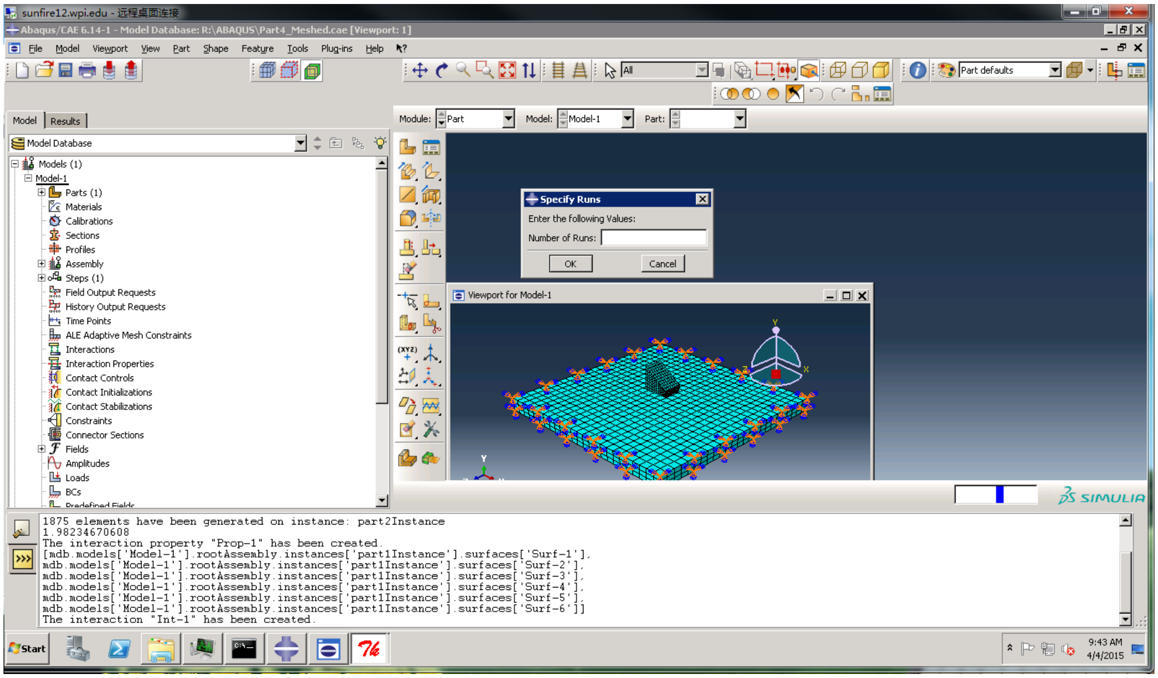
a) Click Randomize Orientation, as shown in Fig.131.


Abaqus/CAE 6.14-1 - Model Database: R:\ABAQUS\Part4_Meshed.cae [Viewport: 1]
Module: Part
Model: Model-1
Viewport for Model-1 Model: Model-1 Step: Initial
Changing Input File
Click this
Randomize Orientation
SIMULIA

```
The model database "R:\ABAQUS\Part4_Meshed.cae" has been opened.
Global seeds have been assigned.
1875 elements have been generated on instance: part2Instance
1.4
The interaction property "Prop-1" has been created.
['6']
[mdb.models['Model-1'].rootAssembly.instances['part1Instance'].surfaces['Surf-1'], mdb.models['Model-1'].rootAssembly.instances['part1Instance'].surfaces['Surf-2'],
mdb.models['Model-1'].rootAssembly.instances['part1Instance'].surfaces['Surf-3'], mdb.models['Model-1'].rootAssembly.instances['part1Instance'].surfaces['Surf-4'],
mdb.models['Model-1'].rootAssembly.instances['part1Instance'].surfaces['Surf-5'], mdb.models['Model-1'].rootAssembly.instances['part1Instance'].surfaces['Surf-6']]
The interaction "Int-1" has been created.
```

Fig. 131. Choose a Method to Modify Input File

i. “Randomize Orientation” – if the user decides to find out the probability distribution of part landing orientation under fixed initial conditions, then only the number of runs is required as input based on the user’s own choice, as shown in Fig. 132.



Module: Part Model: Model-1 Part:

Specify Runs

Enter the following Values:

Number of Runs:

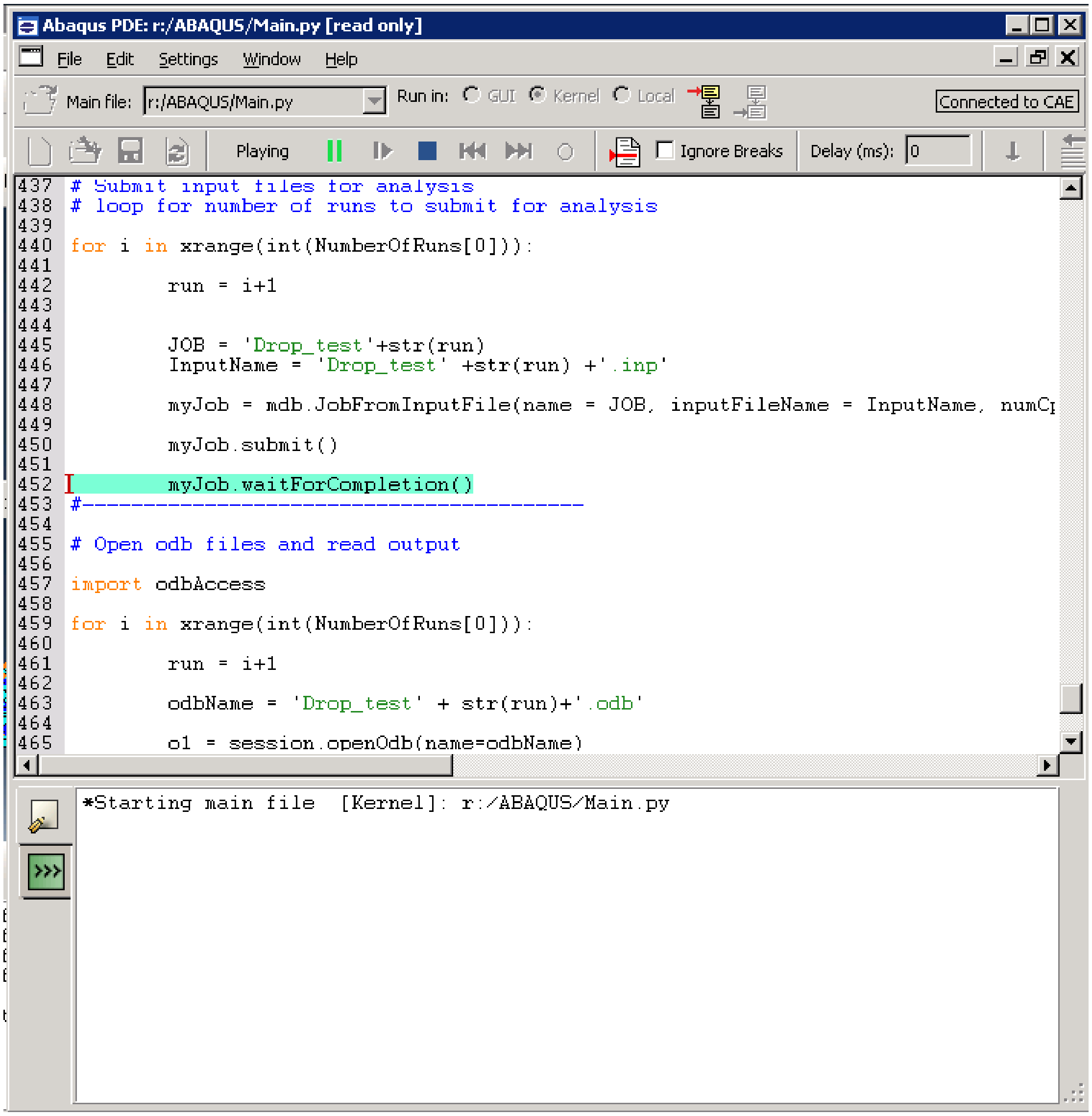
```
1875 elements have been generated on instance: part2Instance
1.98234670608
The interaction property "Prop-1" has been created.
[mdb.models['Model-1'].rootAssembly.instances['part1Instance'].surfaces['Surf-1'],
mdb.models['Model-1'].rootAssembly.instances['part1Instance'].surfaces['Surf-2'],
mdb.models['Model-1'].rootAssembly.instances['part1Instance'].surfaces['Surf-3'],
mdb.models['Model-1'].rootAssembly.instances['part1Instance'].surfaces['Surf-4'],
mdb.models['Model-1'].rootAssembly.instances['part1Instance'].surfaces['Surf-5'],
mdb.models['Model-1'].rootAssembly.instances['part1Instance'].surfaces['Surf-6']]
The interaction "Int-1" has been created.
```

Fig. 132. Randomize Orientation

Until this step, a series of ABAQUS input file is created. All the user inputs are finished and the script will start submitting input files for analysis.

7. Wait for Completion.

   a) If the user goes back to the ABAQUS PDE window, “myJob.waitForCompletion( )” is highlighted as shown in Fig. 133. This means the user needs to wait for all the analysis to be completed.



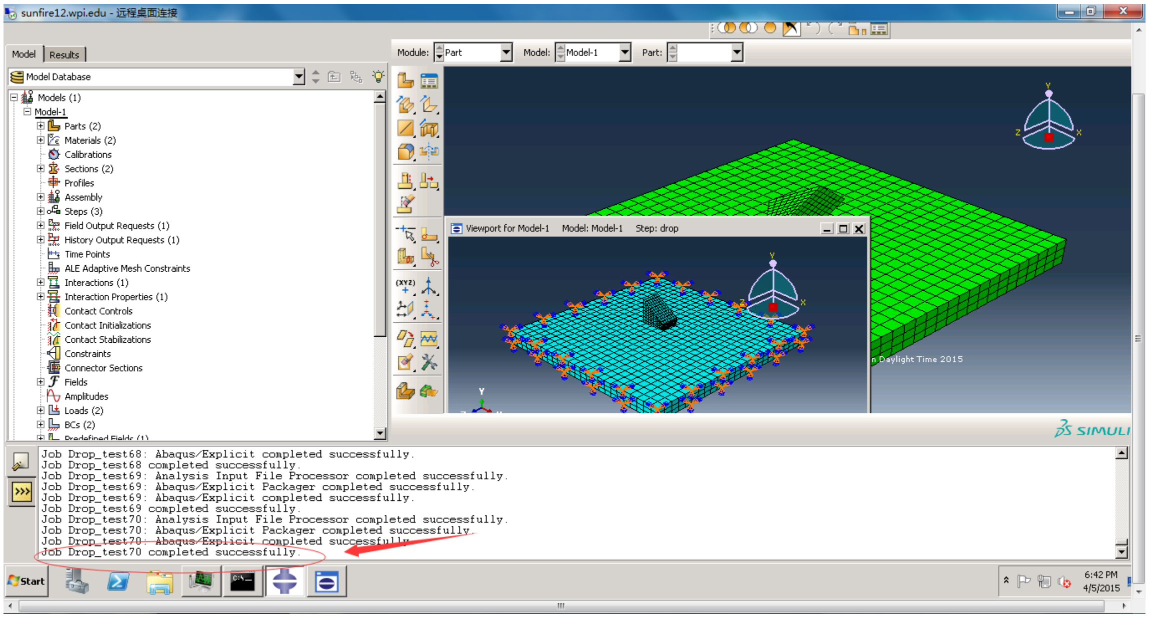
```
# Submit input files for analysis
# loop for number of runs to submit for analysis

for i in xrange(int(NumberOfRuns[0])):

        run = i+1


        JOB = 'Drop_test'+str(run)
        InputName = 'Drop_test' +str(run) +'.inp'

        myJob = mdb.JobFromInputFile(name = JOB, inputFileName = InputName, numC

        myJob.submit()

        myJob.waitForCompletion()
#------------------------------------------

# Open odb files and read output

import odbAccess

for i in xrange(int(NumberOfRuns[0])):

        run = i+1

        odbName = 'Drop_test' + str(run)+'.odb'

        o1 = session.openOdb(name=odbName)
```

```
*Starting main file  [Kernel]: r:/ABAQUS/Main.py
```

Fig. 133. “Wait for completion”

8. Analyses are Done

a) Go back to the ABAQUS GUI command window, as shown in Fig. 134. It will display the status of the running process. For example if the users submitted 70 runs, he/she can tell all the analyses are done by the message “Job Drop_test70 completed successfully”.



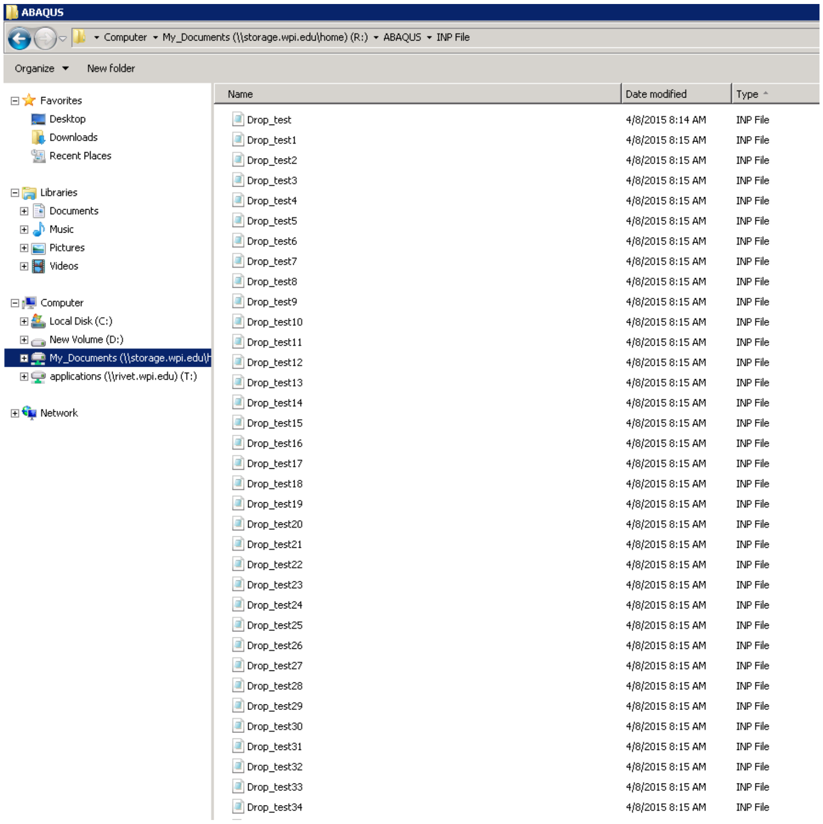
```
Job Drop_test68: Abaqus/Explicit completed successfully.
Job Drop_test68 completed successfully.
Job Drop_test69: Analysis Input File Processor completed successfully.
Job Drop_test69: Abaqus/Explicit Packager completed successfully.
Job Drop_test69: Abaqus/Explicit completed successfully.
Job Drop_test69 completed successfully.
Job Drop_test70: Analysis Input File Processor completed successfully.
Job Drop_test70: Abaqus/Explicit Packager completed successfully.
Job Drop_test70: Abaqus/Explicit completed successfully.
Job Drop_test70 completed successfully.
```

Fig. 134. Analyses are done

9. Collect all the files required for further results interpretation

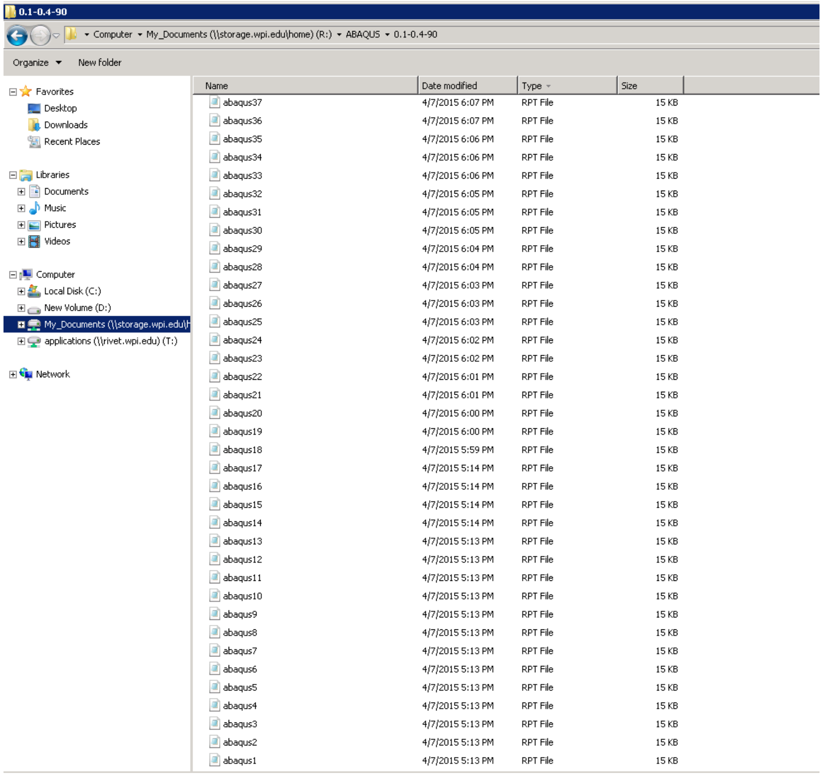
   a) There will be two type of files generated from the analyses in the working directory, one type is INP File which are the input files as shown in Fig. 135, and the other is RPT File which are output files as shown in Fig. 136.

   b) The user needs to collect only one input file and all the output files. For example, for INP type file, only the file titled with “Drop_test” needs to be collected. For RPT type file, all the generated output files need to be collected. If the user specified 70 as the total number of runs, then all the 70 RPT type files need to be collected. Therefore, as in this example, there will be totally 71 files collected.



Name	Date modified	Type
Drop_test	4/8/2015 8:14 AM	INP File
Drop_test1	4/8/2015 8:15 AM	INP File
Drop_test2	4/8/2015 8:15 AM	INP File
Drop_test3	4/8/2015 8:15 AM	INP File
Drop_test4	4/8/2015 8:15 AM	INP File
Drop_test5	4/8/2015 8:15 AM	INP File
Drop_test6	4/8/2015 8:15 AM	INP File
Drop_test7	4/8/2015 8:15 AM	INP File
Drop_test8	4/8/2015 8:15 AM	INP File
Drop_test9	4/8/2015 8:15 AM	INP File
Drop_test10	4/8/2015 8:15 AM	INP File
Drop_test11	4/8/2015 8:15 AM	INP File
Drop_test12	4/8/2015 8:15 AM	INP File
Drop_test13	4/8/2015 8:15 AM	INP File
Drop_test14	4/8/2015 8:15 AM	INP File
Drop_test15	4/8/2015 8:15 AM	INP File
Drop_test16	4/8/2015 8:15 AM	INP File
Drop_test17	4/8/2015 8:15 AM	INP File
Drop_test18	4/8/2015 8:15 AM	INP File
Drop_test19	4/8/2015 8:15 AM	INP File
Drop_test20	4/8/2015 8:15 AM	INP File
Drop_test21	4/8/2015 8:15 AM	INP File
Drop_test22	4/8/2015 8:15 AM	INP File
Drop_test23	4/8/2015 8:15 AM	INP File
Drop_test24	4/8/2015 8:15 AM	INP File
Drop_test25	4/8/2015 8:15 AM	INP File
Drop_test26	4/8/2015 8:15 AM	INP File
Drop_test27	4/8/2015 8:15 AM	INP File
Drop_test28	4/8/2015 8:15 AM	INP File
Drop_test29	4/8/2015 8:15 AM	INP File
Drop_test30	4/8/2015 8:15 AM	INP File
Drop_test31	4/8/2015 8:15 AM	INP File
Drop_test32	4/8/2015 8:15 AM	INP File
Drop_test33	4/8/2015 8:15 AM	INP File
Drop_test34	4/8/2015 8:15 AM	INP File

Fig. 135. INP File

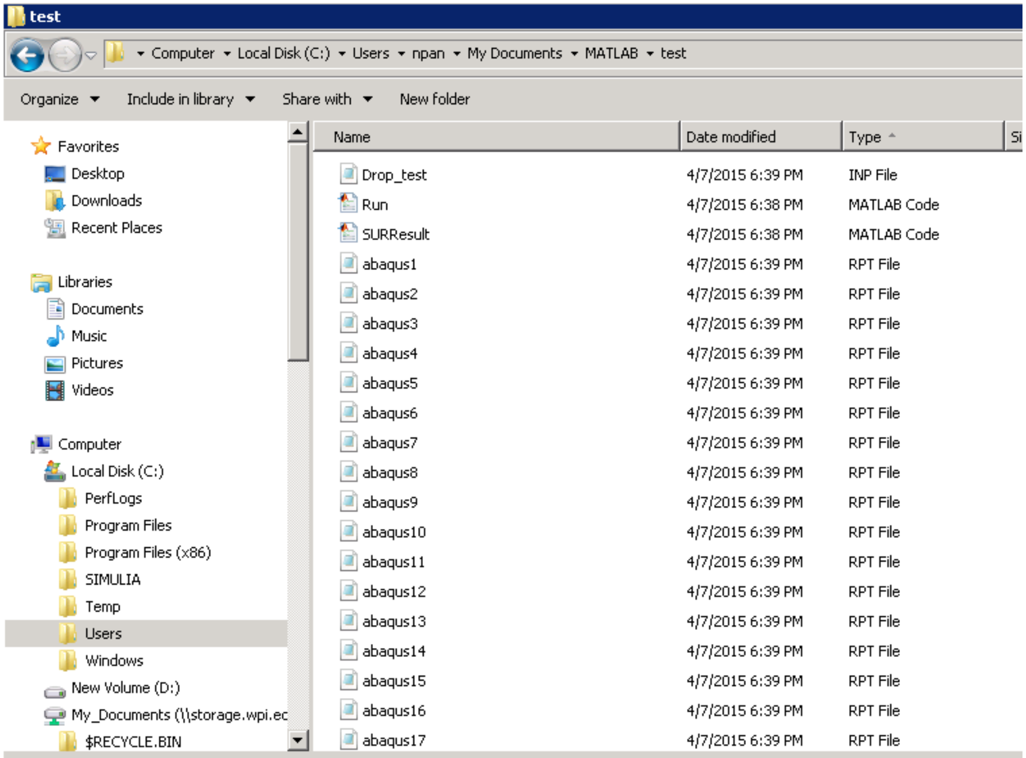


Name	Date modified	Type	Size
abaqus37	4/7/2015 6:07 PM	RPT File	15 KB
abaqus36	4/7/2015 6:07 PM	RPT File	15 KB
abaqus35	4/7/2015 6:06 PM	RPT File	15 KB
abaqus34	4/7/2015 6:06 PM	RPT File	15 KB
abaqus33	4/7/2015 6:06 PM	RPT File	15 KB
abaqus32	4/7/2015 6:05 PM	RPT File	15 KB
abaqus31	4/7/2015 6:05 PM	RPT File	15 KB
abaqus30	4/7/2015 6:05 PM	RPT File	15 KB
abaqus29	4/7/2015 6:04 PM	RPT File	15 KB
abaqus28	4/7/2015 6:04 PM	RPT File	15 KB
abaqus27	4/7/2015 6:03 PM	RPT File	15 KB
abaqus26	4/7/2015 6:03 PM	RPT File	15 KB
abaqus25	4/7/2015 6:03 PM	RPT File	15 KB
abaqus24	4/7/2015 6:02 PM	RPT File	15 KB
abaqus23	4/7/2015 6:02 PM	RPT File	15 KB
abaqus22	4/7/2015 6:01 PM	RPT File	15 KB
abaqus21	4/7/2015 6:01 PM	RPT File	15 KB
abaqus20	4/7/2015 6:00 PM	RPT File	15 KB
abaqus19	4/7/2015 6:00 PM	RPT File	15 KB
abaqus18	4/7/2015 5:59 PM	RPT File	15 KB
abaqus17	4/7/2015 5:14 PM	RPT File	15 KB
abaqus16	4/7/2015 5:14 PM	RPT File	15 KB
abaqus15	4/7/2015 5:14 PM	RPT File	15 KB
abaqus14	4/7/2015 5:14 PM	RPT File	15 KB
abaqus13	4/7/2015 5:13 PM	RPT File	15 KB
abaqus12	4/7/2015 5:13 PM	RPT File	15 KB
abaqus11	4/7/2015 5:13 PM	RPT File	15 KB
abaqus10	4/7/2015 5:13 PM	RPT File	15 KB
abaqus9	4/7/2015 5:13 PM	RPT File	15 KB
abaqus8	4/7/2015 5:13 PM	RPT File	15 KB
abaqus7	4/7/2015 5:13 PM	RPT File	15 KB
abaqus6	4/7/2015 5:13 PM	RPT File	15 KB
abaqus5	4/7/2015 5:13 PM	RPT File	15 KB
abaqus4	4/7/2015 5:13 PM	RPT File	15 KB
abaqus3	4/7/2015 5:13 PM	RPT File	15 KB
abaqus2	4/7/2015 5:13 PM	RPT File	15 KB
abaqus1	4/7/2015 5:13 PM	RPT File	15 KB

Fig. 136. RPT File

10. Put all the collected files into MATLAB working directory

    a) Direct to the working folder of MATLAB, and then copy and paste all the previous collected files into the folder, as shown in Fig.137.



Name	Date modified	Type	Si
Drop_test	4/7/2015 6:39 PM	INP File	
Run	4/7/2015 6:38 PM	MATLAB Code	
SURResult	4/7/2015 6:38 PM	MATLAB Code	
abaqus1	4/7/2015 6:39 PM	RPT File	
abaqus2	4/7/2015 6:39 PM	RPT File	
abaqus3	4/7/2015 6:39 PM	RPT File	
abaqus4	4/7/2015 6:39 PM	RPT File	
abaqus5	4/7/2015 6:39 PM	RPT File	
abaqus6	4/7/2015 6:39 PM	RPT File	
abaqus7	4/7/2015 6:39 PM	RPT File	
abaqus8	4/7/2015 6:39 PM	RPT File	
abaqus9	4/7/2015 6:39 PM	RPT File	
abaqus10	4/7/2015 6:39 PM	RPT File	
abaqus11	4/7/2015 6:39 PM	RPT File	
abaqus12	4/7/2015 6:39 PM	RPT File	
abaqus13	4/7/2015 6:39 PM	RPT File	
abaqus14	4/7/2015 6:39 PM	RPT File	
abaqus15	4/7/2015 6:39 PM	RPT File	
abaqus16	4/7/2015 6:39 PM	RPT File	
abaqus17	4/7/2015 6:39 PM	RPT File	

Fig. 137. MATLAB Working Directory

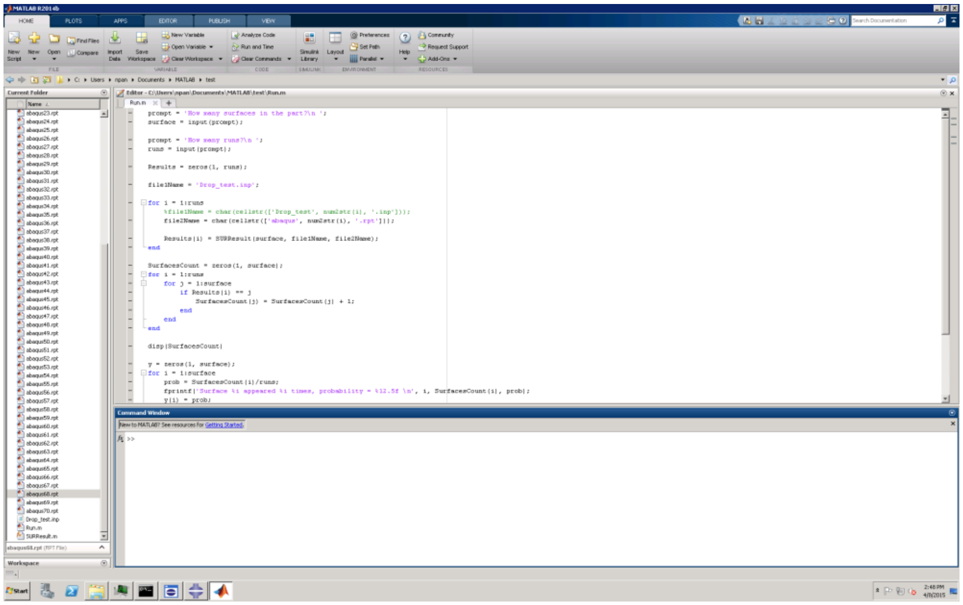
11. Launch MATLAB

a) Double click on MATLAB icon, as shown in Fig.138, to start the MATLAB software

Fig. 138. MATLAB Icon

12. Open MATLAB Script and Run

a) Open the MATLAB script titled with Run. The script will be displayed in the Editor window as shown in Fig. 139.



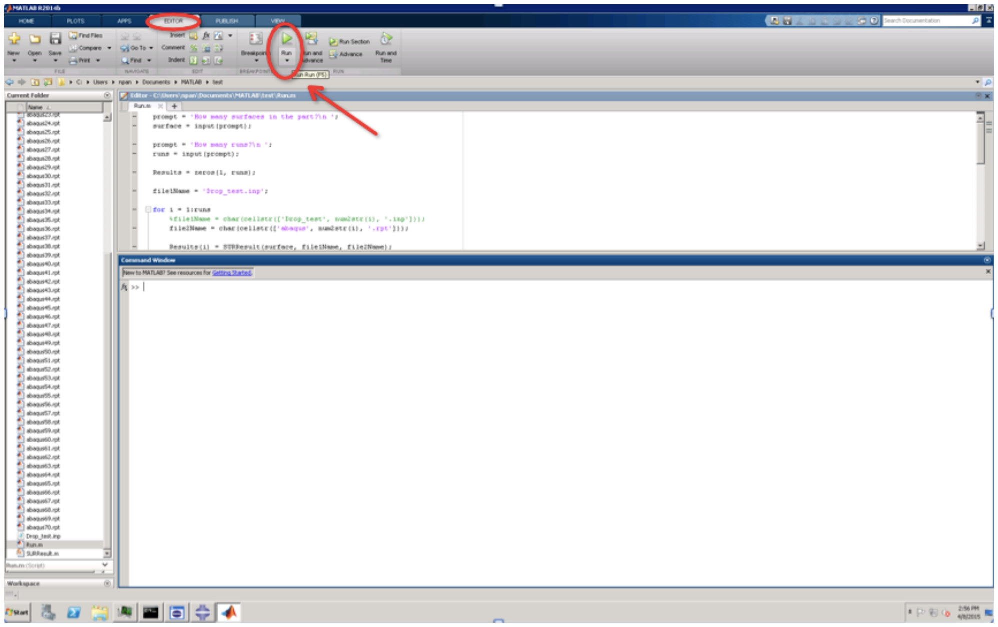
```
prompt = 'How many surfaces in the part?\n ';
surface = input(prompt);

prompt = 'How many runs?\n ';
runs = input(prompt);

Results = zeros(1, runs);

file1Name = 'Drop_test.inp';

for i = 1:runs
    %file1Name = char(cellstr(['Drop_test', num2str(i), '.inp']));
    file2Name = char(cellstr(['abaqus', num2str(i), '.rpt']));

    Results(i) = SURResult(surface, file1Name, file2Name);
end

SurfacesCount = zeros(1, surface);
for i = 1:runs
    for j = 1:surface
        if Results(i) == j
            SurfacesCount(j) = SurfacesCount(j) + 1;
        end
    end
end

disp(SurfacesCount)

y = zeros(1, surface);
for i = 1:surface
    prob = SurfacesCount(i)/runs;
    fprintf('Surface %i appeared %i times, probability = %12.5f \n', i, SurfacesCount(i), prob);
    y(i) = prob;
```

New to MATLAB? See resources for Getting Started.

```
>>
```

Fig. 139. Open MATLAB Script

b) Click the Editor Tab on the Top and then Click Run to start running the script, as shown in Fig. 140.



```
prompt = 'How many surfaces in the part?\n ';
surface = input(prompt);

prompt = 'How many runs?\n ';
runs = input(prompt);

Results = zeros(1, runs);

file1Name = 'Drop_test.inp';

for i = 1:runs
    %file1Name = char(cellstr(['Drop_test', num2str(i), '.inp']));
    file2Name = char(cellstr(['abaqus', num2str(i), '.rpt']));

    Results(i) = STRResult(surface, file1Name, file2Name);
```

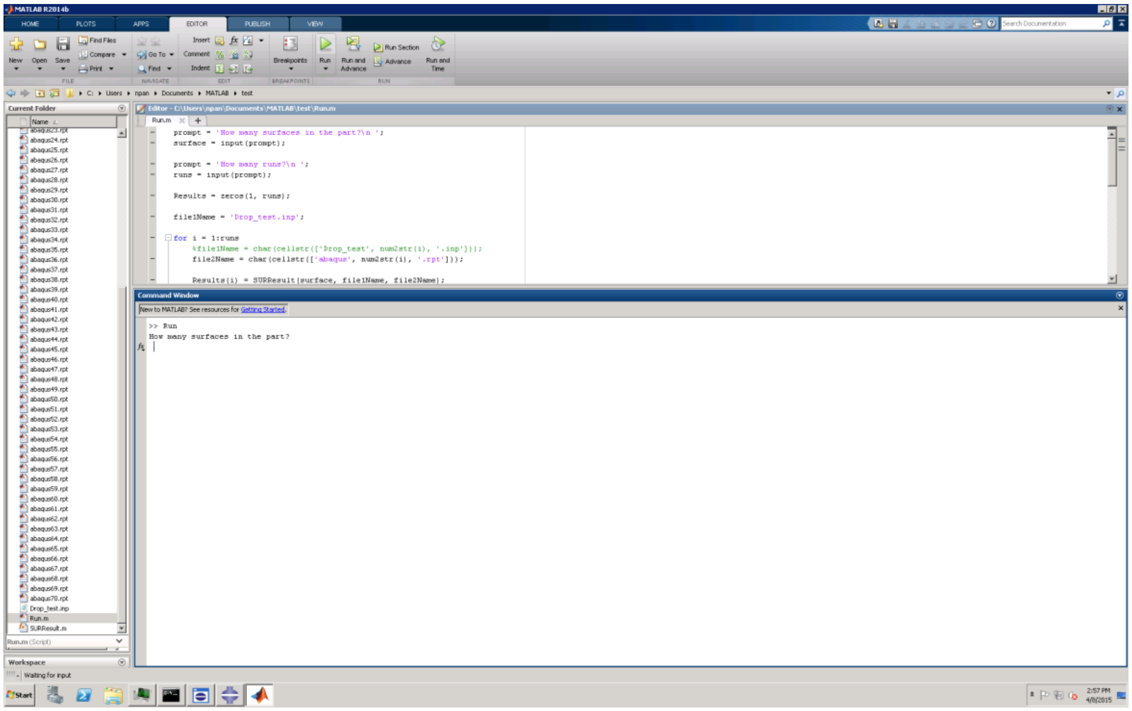
```
>>
```

Fig. 140. Run MATLAB Script

13. Enter inputs

a) How many surfaces in the part?

i. The user needs to enter an integer value, which is the number of labeled contact surfaces in the dropping model. This number should be consistent with the number input in step 5(d) during the ABAQUS simulation. The user interface is shown in Fig. 141. For example: enter 6 and then press Enter key.



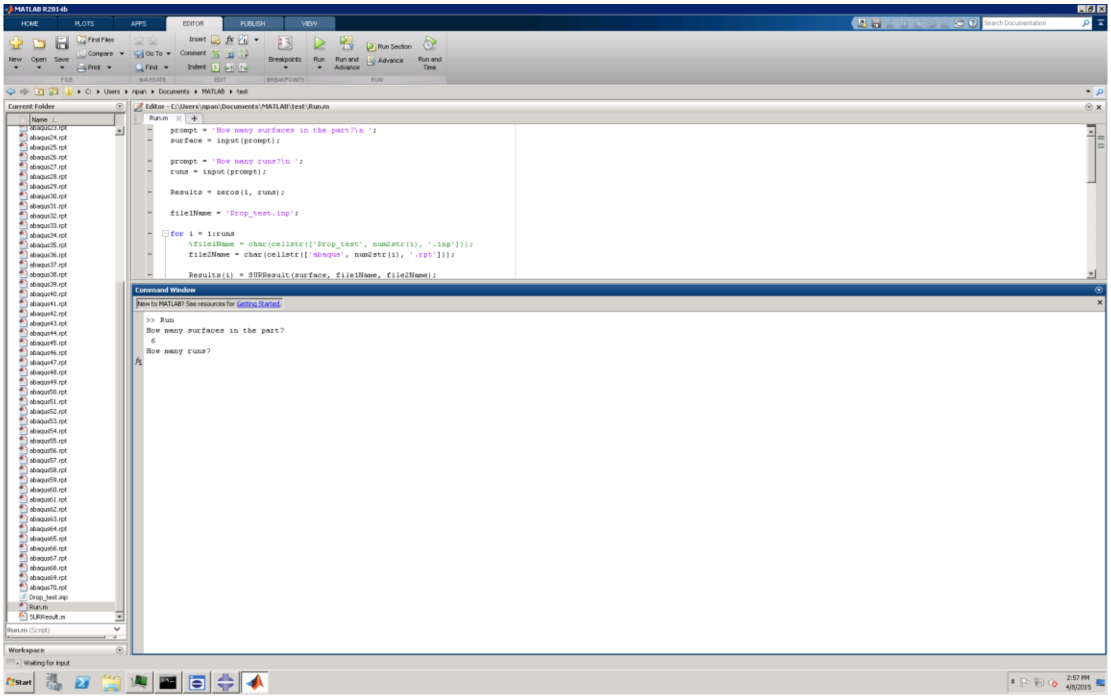
```
prompt = 'How many surfaces in the part?\n ';
surface = input(prompt);

prompt = 'How many runs?\n ';
runs = input(prompt);

Results = zeros(1, runs);

file1Name = 'Drop_test.inp';

for i = 1:runs
    %file1Name = char(cellstr(['Drop_test', num2str(i), '.inp']));
    file2Name = char(cellstr(['abaqus', num2str(i), '.rpt']));

    Results(i) = SURResult(surface, file1Name, file2Name);
```

```
>> Run
How many surfaces in the part?
```

Fig. 141. Input Number of Surfaces

b) How many runs?

  i. The user needs to enter an integer, which is the total number of runs of simulation that are done in ABAQUS. This also indicates the total number of output files available in the MATLAB working directory. The user interface is shown in Fig. 142. For example: enter 70 and then press Enter key.

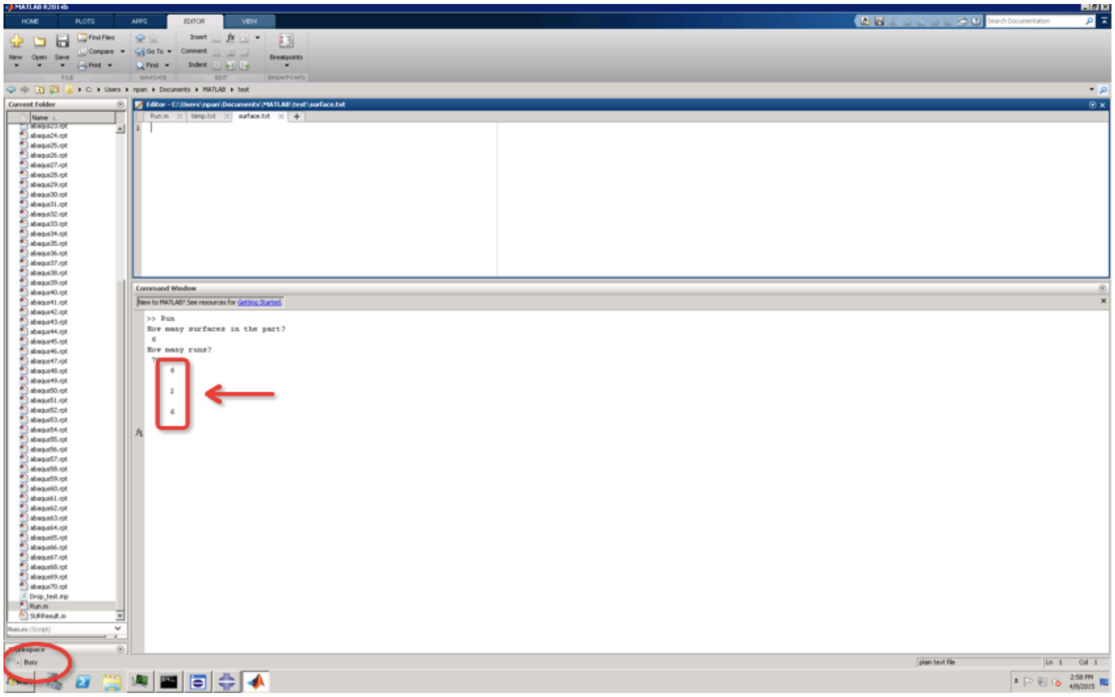


```
prompt = 'How many surfaces in the part?\n ';
surface = input(prompt);

prompt = 'How many runs?\n ';
runs = input(prompt);

Results = zeros(1, runs);

file1Name = 'Drop_test.inp';

for i = 1:runs
    %file1Name = char(cellstr(['Drop_test', num2str(i), '.inp']));
    file2Name = char(cellstr(['abaqus', num2str(i), '.rpt']));

    Results(i) = SURResult(surface, file1Name, file2Name);
```

```
>> Run
How many surfaces in the part?
 6
How many runs?
```

Fig. 142. Input Number of Runs

c) After all the inputs are entered, the MATLAB script will start to display the results in the command window as shown in Fig. 143. The lower left corner indicates the status of the process. The script is still running when it show “Busy”.

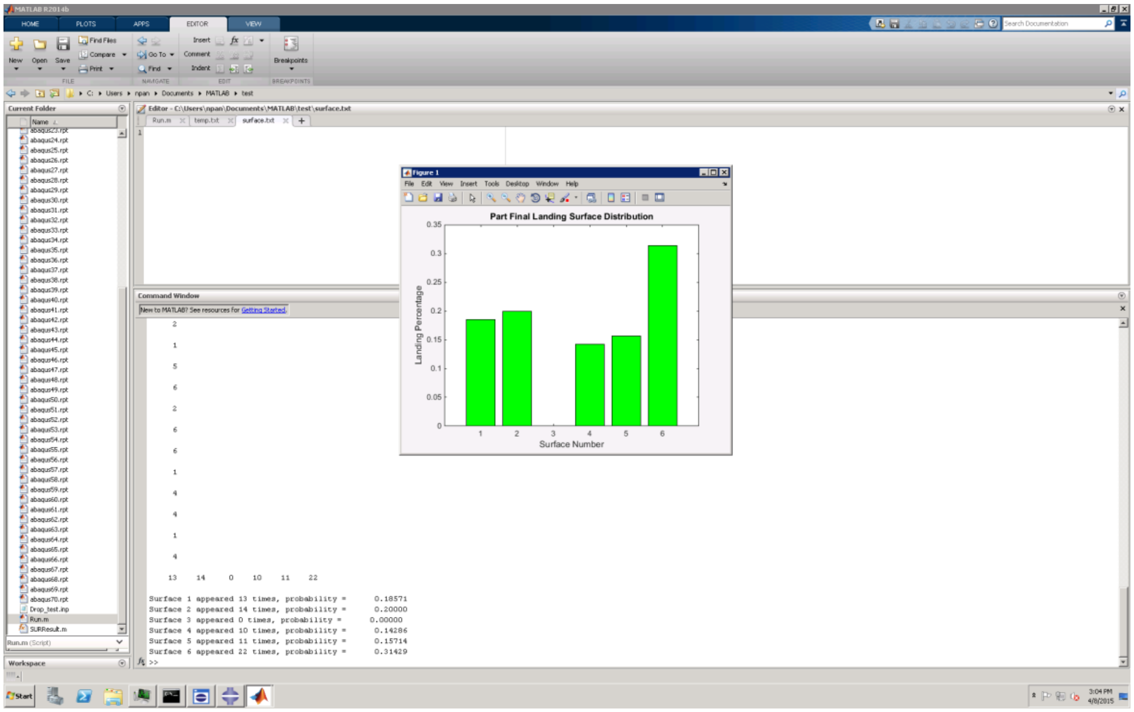


```
New to MATLAB? See resources for Getting Started.
>> Run
How many surfaces in the part?
 6
How many runs?
 7
      6

      2

      6
```

Fig. 143. MATLAB Script Running

14. Obtain Final Simulation Results.

a) After the script is finished running, a histogram indicates the probability distribution of part landing orientation and a text message of detailed information will be shown as in Fig. 144.



```
     2

     1

     5

     6

     2

     6

     6

     1

     4

     4

     1

     4

    13    14     0    10    11    22

Surface 1 appeared 13 times, probability =      0.18571
Surface 2 appeared 14 times, probability =      0.20000
Surface 3 appeared 0 times, probability =      0.00000
Surface 4 appeared 10 times, probability =      0.14286
Surface 5 appeared 11 times, probability =      0.15714
Surface 6 appeared 22 times, probability =      0.31429
>>
```

Fig. 144. MATLAB Script Finished

This concludes the instructions.

Appendix P. Additional Data from High-speed Footage

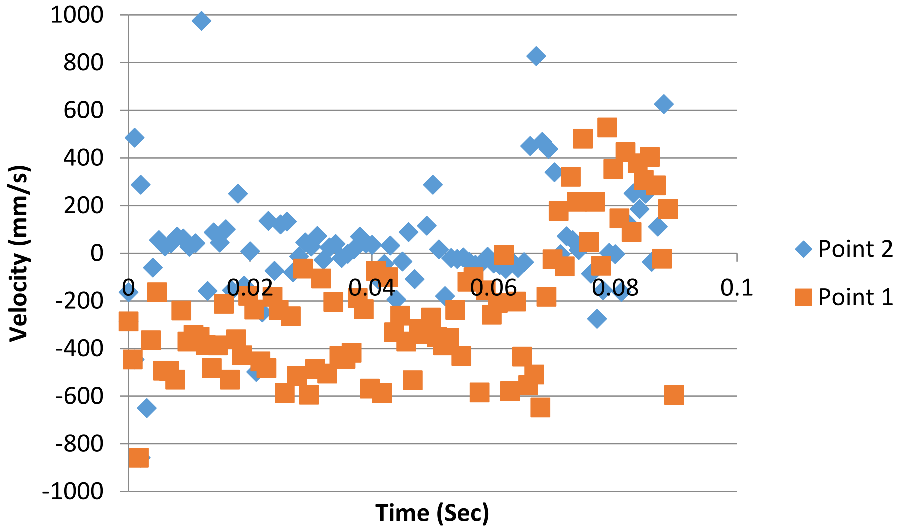


Fig 145. Graphical representation of velocities in the x direction of Control Points 1 and 2 during Test 1, where the part lands on its rigid surface.

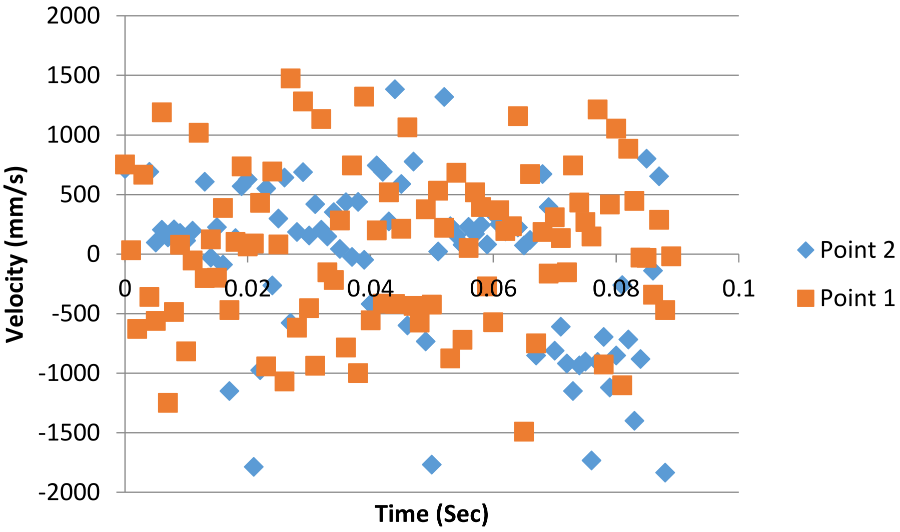


Fig 146. Graphical representation of velocities in the z direction of Control Points 1 and 2 during Test 1, where the part lands on its rigid surface.

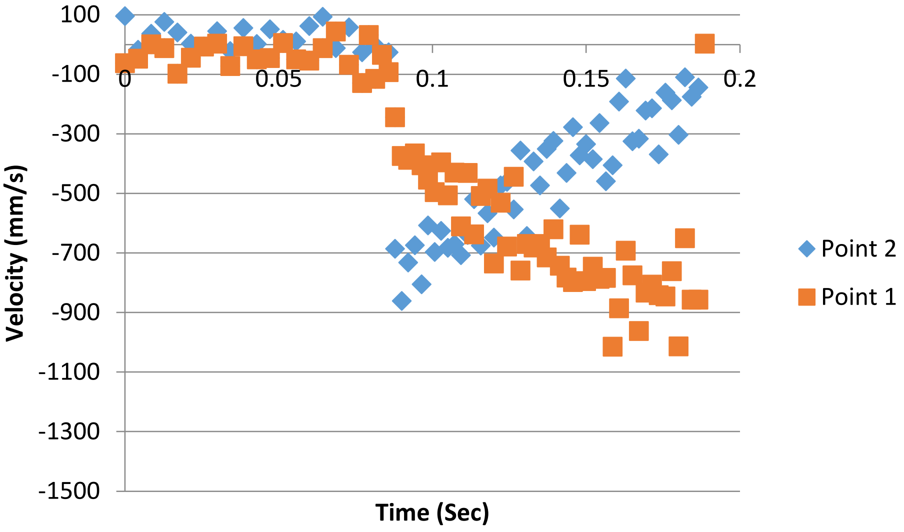


Fig 147. Graphical representation of velocities in the x direction of Control Points 1 and 2 during Test 2, where the part lands on its elastic surface.

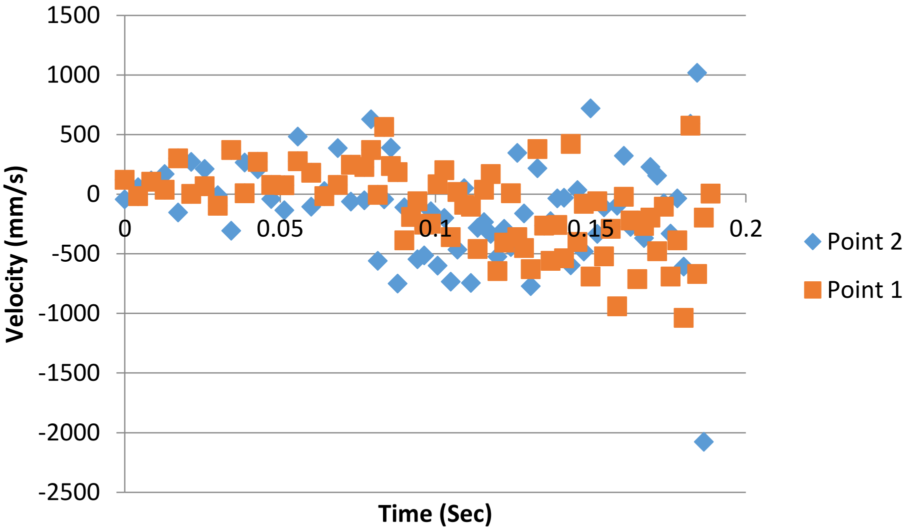


Fig 148. Graphical representation of velocities in the z direction of Control Points 1 and 2 during Test 2, where the part lands on its elastic surface.

Appendix Q. Variable Dependency Analysis

FEA Analyses Results

To investigate relationship between varying velocity values and the part final landing orientation, analyses are done under controlled height value of 0.2m and angle of 90 degree. The velocity is varied at 0.1, 0.2 and 0.3 meter per second. The observed FEA analyses results are shown in Table 26.

Table 26. Analyses Results for Velocity

Observed Landing Frequency Under Different Velocity Values				
Landing Orientation	0.1	0.2	0.3	Total
Surf-1	27	32	34	93
Surf-2	11	10	4	25
Surf-3	21	13	16	50
Surf-4	11	15	16	42
Total	70	70	70	210

To investigate relationship between varying height values and the part final landing orientation, analyses are done under controlled velocity value of 0.1m/s and angle of 90 degree. The height is varied at 0.2, 0.3 and 0.4 meter. The observed FEA analyses results are shown in Table 27.

Table 27. Analyses Results for Height

Observed Landing Frequency Under Different Height Values				
Landing Orientation	0.2	0.3	0.4	Total
Surf-1	27	28	23	78
Surf-2	11	17	22	50
Surf-3	21	10	14	45
Surf-4	11	15	11	37
Total	70	70	70	210

To investigate relationship between varying angle values and the part final landing orientation, analyses are done under controlled velocity value of 0.1m/s and height of 0.2m. The angle is varied at 90, 80 and 70 degrees. The observed FEA analyses results are shown in Table 28.

Table 28. Analyses Results for Angle

Observed Landing Frequency Under Different Angle Values				
Landing Orientation	90	80	70	Total
Surf-1	27	39	31	97
Surf-2	11	7	10	28
Surf-3	21	13	16	50
Surf-4	11	11	13	35
Total	70	70	70	210

Chi Square Test Procedure and Results

Chi square test is used to determine whether there is a significant association between the two variables. Here in our application, chi square test is used to determine whether there is a significant association between each initial dropping variable (velocity, height and shoot angle) and the part final landing orientations.

For each initial dropping variable there are 3 levels and for the landing orientation variable there are 4 levels. The null hypothesis in this test is that the initial dropping variable and the landing orientation variable are independent. The degrees of freedom used in this test is 6, which is calculated from the product of the number of columns minus one and the number of rows minus one.

Expected frequency counts are calculated separately for each level of one categorical variable at each level of the other categorical variable. The corresponding expected values for each initial dropping variable are shown in Table 29, 30, and 31.

Table 29. Expected Values for Velocity

Expected Landing Frequency Under Different Velocity Values			
Landing Orientation	0.1	0.2	0.3
Surf-1	31	31	31
Surf-2	8.33	8.33	8.33
Surf-3	16.67	16.67	16.67
Surf-4	14	14	14

Table 30. Expected Values for Height

Expected Landing Frequency Under Different Height Values			
Landing Orientation	0.2	0.3	0.4
Surf-1	26	26	26
Surf-2	16.67	16.67	16.67
Surf-3	15	15	15
Surf-4	12.33	12.33	12.33

Table 31. Expected Values for Angle

Expected Landing Frequency Under Different Angle Values			
Landing Orientation	90	80	70
Surf-1	32.33	32.33	32.33
Surf-2	9.33	9.33	9.33
Surf-3	16.67	16.67	16.67
Surf-4	11.67	11.67	11.67

The method used for calculating chi square test value and p-value can be found in Appendix E. The corresponding results of chi square test value and p-value for each initial dropping variable can be found in Table 32, 33 and 34.

Table 32. Test Results for Velocity

Chi Square Test Value and P-Value Calculation For Velocity				
Observed (O)	Expected (E)	O-E	(O-E)^2	(O-E)^2/E
27	31	-4	16	0.516
32	31	1	1	0.032
34	31	3	9	0.290
11	8.33	2.67	7.129	0.856
10	8.33	1.67	2.789	0.335
4	8.33	-4.33	18.749	2.251
21	16.67	4.33	18.749	1.125
13	16.67	-3.67	13.469	0.808
16	16.67	-0.67	0.449	0.027
11	14	-3	9	0.643
15	14	1	1	0.071
16	14	2	4	0.286
Chi Square Test Value				7.240
P - Value				0.299

Table 33. Test Results for Height

Chi Square Test Value and P-Value Calculation For Height				
Observed (O)	Expected (E)	O-E	(O-E)^2	(O-E)^2/E
27	26	1	1	0.038
28	26	2	4	0.154
23	26	-3	9	0.346
11	16.67	-5.67	32.149	1.929
17	16.67	0.33	0.109	0.007
22	16.67	5.33	28.409	1.704
21	15	6	36	2.400
10	15	-5	25	1.667
14	15	-1	1	0.067
11	12.33	-1.33	1.769	0.143
15	12.33	2.67	7.129	0.578
11	12.33	-1.33	1.769	0.143
	Chi Square Test Value			9.176
	P - Value			0.164

Table 34. Test Results for Angle

Chi Square Test Value and P-Value Calculation For Angle				
Observed (O)	Expected (E)	O-E	(O-E)^2	(O-E)^2/E
27	32.33	-5.33	28.409	0.879
39	32.33	6.67	44.489	1.376
31	32.33	-1.33	1.769	0.055
11	9.33	1.67	2.789	0.299
7	9.33	-2.33	5.429	0.582
10	9.33	0.67	0.449	0.048
21	16.67	4.33	18.749	1.125
13	16.67	-3.67	13.469	0.808
16	16.67	-0.67	0.449	0.027
11	11.67	-0.67	0.449	0.038
11	11.67	-0.67	0.449	0.038
13	11.67	1.33	1.769	0.152
	Chi Square Test Value			5.427
	P - Value			0.49