

# PERFORMING A SIMPLE PENDULUM EXPERIMENT AS A DEMONSTRATIVE EXAMPLE FOR COMPUTER EXPERIMENTS

Kazeem A. Osulale, Waheed B. Yahya, Babatunde L. Adeleke

Department of Statistics, University of Ilorin, P.M.B 1515, Ilorin, Nigeria

**ABSTRACT:** A computer experiment is an experiment conducted using data obtained from a computer model or simulator in lieu of the physical process. A physics-based experiment known as a simple pendulum experiment was performed as a demonstrative example for a computer experiment. A true model called a simulator or a computer model of a simple pendulum was used to mimic a real life pendulum experiment. The inputs to the computer code were varied in order to determine the effect of different inputs on the output(s). The outputs of such computer model or simulator served as a proxy for the real life observations of the study. Our interest in this study is to determine the output which is the time it takes the bob of a pendulum to return to rest. This time is also called the stoppage time in this research. MATLAB 2012a computer package ([www.mathworks.com/](http://www.mathworks.com/)) was used for the development of the program that generates the time it takes the pendulum to return to rest.

**KEYWORDS:** Computer experiment, Simulator, Simple Pendulum, output, Stoppage time

## 1. INTRODUCTION

Deterministic computer experiments are becoming more commonly used in science and engineering. This is mainly because the underlying physical experiments are too time consuming, expensive, or even impossible to perform. The rapid growth in computer power has made it possible to perform deterministic experiments on simulators.

Since the emergence of the first computer experiment conducted by Enrico Fermi and colleagues, Strogatz [Str03] in Los Alamos in 1953, scientists in diverse areas such as engineering, cosmology, particle physics and aircraft design have turned to computer experiments as a powerful tool to understand their respective processes. For instance, in the design of a vehicle, computer experiments are used to study the effect of a collision of the vehicle with a barrier before manufacturing the prototype of the vehicle, see Bayarri [B+02].

A simple pendulum is used in this work as a demonstrative example for a computer experiment. A simple pendulum may be described ideally as a point mass suspended by a weightless string from

some point about which it is allowed to swing back and forth in a place, Parks [Par00]. A simple pendulum can be approximated by a small metal sphere which has a small radius and a large mass when compared relatively to the length and mass of the light string from which it is suspended. The goal of this research is to perform a simple pendulum experiment as a demonstrative example for a computer experiment. The MATLAB code was written to numerically solve for the time the pendulum stops by iteratively varying the pendulum length and initial displacement angle and calculate the time it takes for the pendulum to stop.

## 2. MATERIALS AND METHODS

### 2.1 Brief Overview of Computer Experiments

An *experimental design* is defined as a matrix of input variable values ( $X$ ), where each column of  $X$  represents a variable and each row represents the combination of input variable values for a single run. Classical experimental designs according to Montgomery [Mon01] originated from the theory of Design of Experiments when physical experiments are performed. Computer experiments are different from physical experiments in that they have no random error and they deal with functions that are thought to have more complex behaviour. Properly designed experiments are essential for effective computer utilization.

After the computer experimentation had been carried out based on the simulated model, the next step is to choose a surrogate or approximate model to fit the process. Motulsky [MC03] reported G.E.P. Box to have said that “all models are wrong but some are useful” while many models and methods still exist. A complex mathematical model that produces a set of output values given a set of input values is commonly referred to as a computer model. This model is used to mimic the real life system or physical experiment. More often than not, we want computers to do the extensive computations especially for situations where the intending model does not result to a closed form solution or the one that requires an iterative

solution. Computer models are distinct from models of data from physical experiments in that they are often not subjected to random error. In a computer experiment, observations are made on a response function  $y$  by running a complex computer model at various settings of input factors  $X$ . For instance, in the proposed simple pendulum experiment here,  $X$  can be a set of initial displacement angles in a system of differential equation and  $y$  can be the time it takes the pendulum to return to rest. Solving the resulting differential equations numerically for specified  $X$  gives a value for  $y$ . The goal of this model is to estimate the relationship between  $X$  and  $y$  from a moderate number of runs so that  $y$  can be predicted at untried inputs. Observations obtained from this experiment can be used to build up a computationally cheap surrogate model to the selected simulator or computer code. This surrogate model is used to approximate the computer simulator or model efficiently and inexpensively to investigate the behaviour of the function. A computer always produces the same output when it is fed with the same input. Due to the lack of random error, classical statistical modelling approaches are not appropriate. For example, one of the features of design of experiments is replication which simply results to redundant information in computer experiments. A major contribution to this area was made by Sacks [SSW89]. In many scientific investigations, complex physical phenomena are represented by a mathematical model

$$Y = f(X), X \in [0,1]^m, \quad (1)$$

where  $X$  consists of  $m$  input variables,  $f$  is the functional form of the relationship (computer code) and  $Y$  represents the response. This model is usually a solution to a set of equations, which can be linear, nonlinear, ordinary or partial differential in nature. Due to the complex nature of  $f(X)$  for many real life cases, the solution to equation (1) is often impossible to obtain analytically. Therefore, the scientists often study the complex relationship between the inputs  $X$  and outputs  $y$  by varying the inputs to the computer code and observing how the process outputs are affected.

## 2.2. Simple Pendulum Computer Model

A simple pendulum may be described ideally as a point mass suspended by a massless string from some point about which it is allowed to swing back and forth in a plane, Parks [Par00]. A simple pendulum can be approximated by a small metal sphere which has a small radius and a large mass when compared relatively to the length and mass of

the light string from which it is suspended. If a pendulum is set in motion so that it swings back and forth, its motion will be periodic. The time that it takes to make one complete oscillation is defined as the period  $T$ . The pendulum has the following non-linear differential equation as its model:

$$ML^2 \frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} + MgL \sin(\theta) = 0$$

$$\frac{d^2\theta}{dt^2} = \frac{-b \frac{d\theta}{dt} - MgL \sin(\theta)}{ML^2} \quad (2)$$

where  $M$  is the mass (in  $kg$ ) of the bob,  $L$  is the length (in *metre*) of the pendulum,  $\theta$  is the pendulum displacement angle (in degrees),  $b$  is the coefficient of friction (in *Newton*) and  $g$  is the acceleration (in  $ms^{-2}$ ) due to gravity. The MATLAB code was written to numerically solve for the time the pendulum stops by iteratively varying the pendulum length and initial displacement angle and calculate the time it takes for the pendulum to stop. This model represents a system with two inputs (pendulum length and initial displacement angle) and one output (the stoppage time of the pendulum). The mass,  $M$  is specified while the friction ( $b$ ) is also fixed. The model can be diagrammatically represented as shown below in Figure 1.

## 2.3. Implementation

A computer program using MATLAB 2012a package was written to generate values for the input variables and concurrently calculate the output values. The acceleration due to gravity  $g$  was fixed at  $9.81 \text{ ms}^{-2}$  mass was fixed at  $0.1 \text{ kg}$  and the friction,  $b$  was set to be  $0.25 \text{ N}$ . We built up a complete diagrammatic model for a simple pendulum experiment to represent the non-linear differential equations used in this study. This diagram is shown in Figure 2 to aid the understanding of the input and output values generated from this experiment.

The simulink environment visually solves the model using a chosen solver (ode45) to numerically compute the value of  $\theta$  (pendulum displacement angle). The simulation is run infinitely until the angular displacement becomes zero and the time it occurs determined. The program loads the simulink model and passes the input parameters into the model and returns the stoppage time during each experimental iteration. The Pendulum Length  $L$  was varied between  $0.2 \text{ m}$  and  $1 \text{ m}$  while the initial displacement angle  $\theta$  was varied between  $5^\circ$  and

180° at regular increment of 5° intervals. The results from this experiment are presented in Table 1.

### 3.0 Results and Discussion

This section presents the simulation results as shown by the line graphs in Figure 3. These graphs are presented in order to see at glance, the time the pendulum stops at different degrees of displacement angle when the pendulum length L is 0.2m, 0.4m, 0.6m, 0.8m and 1m. The line graphs showed separately the time it took the pendulum to stop at each pendulum length used in the experiment. The minimum time it took the pendulum to return to rest was 0.2s at 180° and the average displacement angle using harmonic mean was 43.12° for all the pendulum lengths used. The maximum time it took the pendulum to return to rest when L= 0.2m is 8s at 165° and the average time it took the pendulum to stop using harmonic mean was 3.38s. It took the

pendulum the maximum time of 7.64s to return to rest when L= 0.4m at 95° and the average time was 2.71s. The pendulum spent the maximum time of 7.08s to stop when L=0.6m at 135° and the average time was 3.30s. The pendulum also spent the maximum time of 12.83s and 19s to stop when L=0.8m at 165° and L=1.0m at 150° and the average times were 4.38s and 5.03s respectively.

We also used the box plot as displayed in Figure 4 to compress the information contained in Figure 3. This enables easy comparisons of the median stoppage times at different chosen pendulum lengths. From the boxplots, it can be observed that the minimum and maximum stoppage times of the pendulum at all the displacement angles were attained at the pendulum lengths of 0.4m and 1.0m respectively.

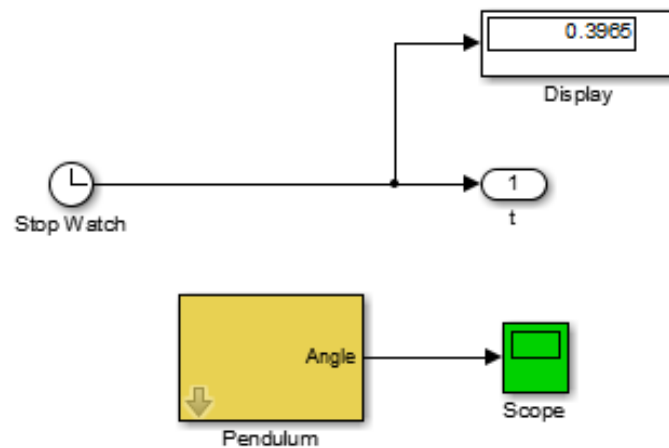


Figure 1: Simple Pendulum model

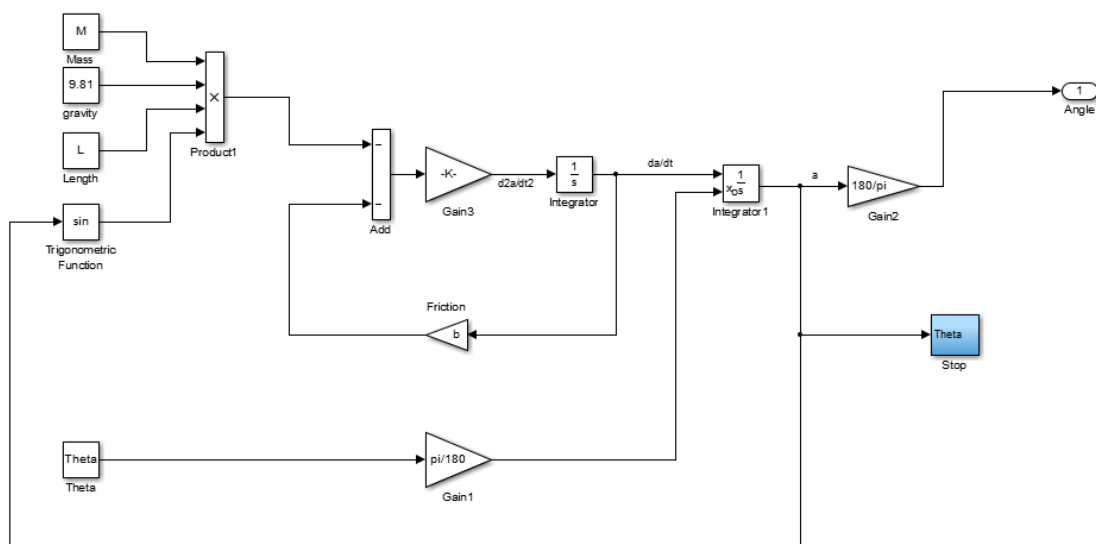


Figure 2: Simple Pendulum model build-up

**Table 1: The estimated stoppage time of the pendulum (in seconds) at various displacement angles for five different pendulum lengths**

<b>Initial Displacement Angle</b>	<b>Stoppage Time at Pendulum length of 0.2</b>	<b>Stoppage Time at Pendulum length of 0.4</b>	<b>Stoppage Time at Pendulum length of 0.6</b>	<b>Stoppage Time at Pendulum length of 0.8</b>	<b>Stoppage Time at Pendulum length of 1.0</b>
5	5.19	3.40	4.27	7.49	10.28
10	5.98	5.85	4.25	8.41	11.28
15	4.46	4.23	5.11	8.41	12.32
20	5.79	1.77	5.10	10.25	14.35
25	5.03	3.42	5.93	9.34	14.35
30	5.40	5.05	5.11	9.35	14.35
35	5.97	5.86	5.94	10.34	14.37
40	5.36	6.68	5.93	10.28	15.40
45	5.73	7.53	5.12	10.29	15.40
50	6.16	5.88	5.11	10.29	16.43
55	5.55	6.71	5.11	10.29	15.43
60	6.98	6.72	5.95	10.32	16.45
65	5.50	6.74	6.81	10.31	15.44
70	7.06	5.12	6.82	12.18	15.45
75	5.58	3.48	6.00	10.34	15.48
80	6.83	2.69	6.01	11.36	18.55
85	5.67	2.69	6.85	12.21	16.52
90	6.91	4.35	6.02	11.32	17.56
95	5.75	7.64	6.03	11.33	15.55
100	7.31	5.20	6.90	11.34	17.60
105	5.83	3.58	6.09	11.44	16.61
110	6.66	2.80	6.11	11.47	18.68
115	7.17	2.82	6.11	12.33	18.70
120	6.01	5.31	6.15	11.50	17.72
125	7.25	6.98	6.17	11.46	18.77
130	6.09	4.55	6.19	12.41	18.81
135	6.61	2.96	7.08	12.46	18.85
140	7.45	2.98	6.27	11.56	18.88
145	6.42	6.31	6.32	12.54	18.93
150	6.39	6.38	6.38	12.58	19.00
155	6.50	3.99	6.41	11.73	18.05
160	6.61	3.25	6.52	12.73	18.13
165	8.00	6.62	6.61	12.83	18.23
170	7.00	4.33	6.73	12.11	18.39
175	7.83	4.57	7.00	12.33	18.64
180	0.20	0.20	0.20	0.20	0.20

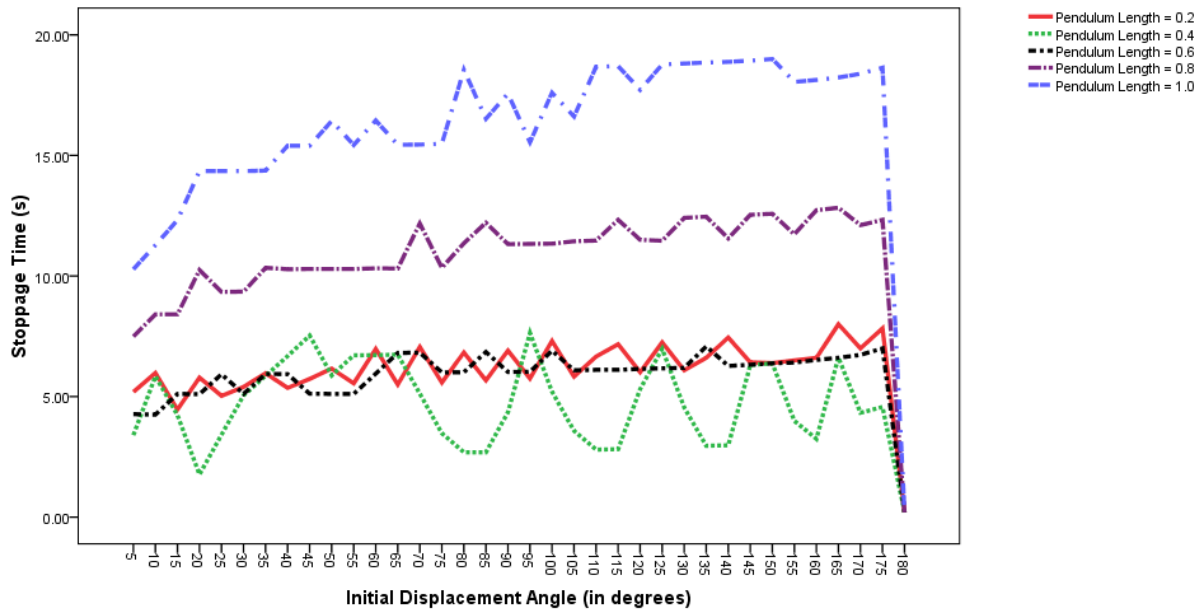


Figure 3: Line graphs of the Stoppage Time against initial displacement angles (in degrees) for five different pendulum lengths

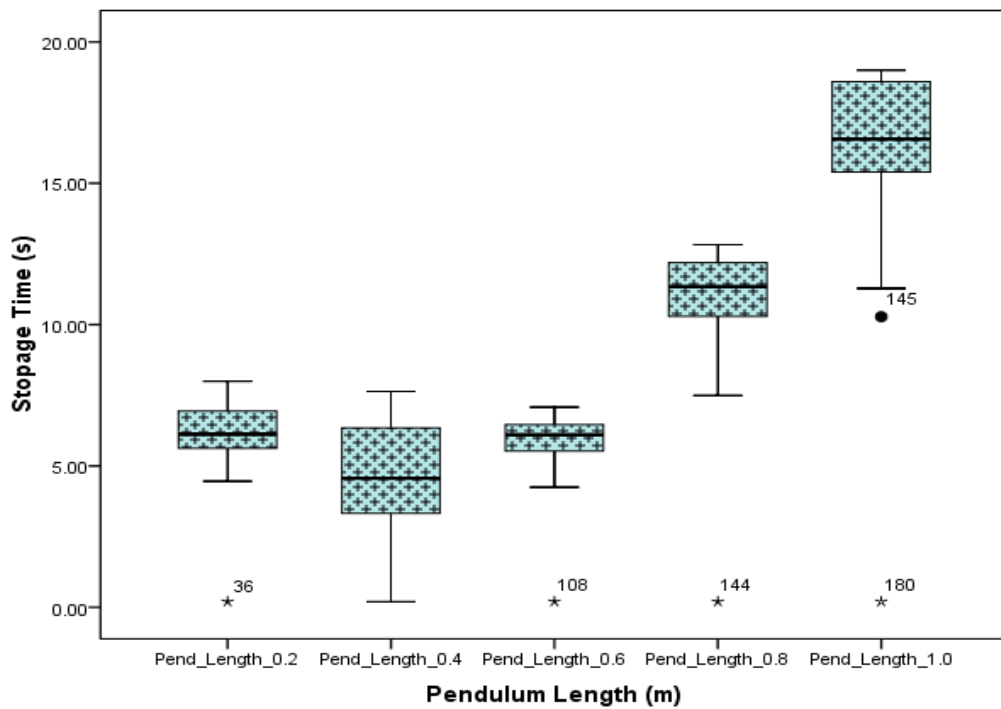


Figure 4: Box Plots of the Stoppage Times at the five chosen Pendulum Lengths over all the displacement angles

```

function [M,MM,dataset_result,D,L] = simple_pendulum_sim(criteria,trials)
1
2
3 M=[];
4 MM=[];
5 warning off
6
7 if nargin == 0
8     criteria = 'maximin';
9     trials = 50;
10 elseif nargin == 1
11     trials = 50;
12 end
13
14 %% Open model
15 load_system('simple_pendulum')
16
17 % Parameters
18 div = 49;
19 %Len = [0.2 0.4 0.6 0.8 1];
20 Len = linspace(0.2,1,div)';
21 %Ang = 0.5:180;
22 Ang = floor(linspace(5,90,div))';
23 %Mass = 0.1;
24 Mass = linspace(0.1,div)';
25
26 Twithout = zeros(1,trials);
27 Twith = zeros(1,trials);
28 cnt = zeros(1,trials);
29 for tt = 1:trials
30     % Change dialog
31     cdc
32     fprintf('Performing trial %d out of %d\n',tt,trials)
33     cnt(tt) = tt;

```

Figure 5. MATLAB Code for the simulation of a simple pendulum model

## CONCLUSIONS

This paper presents the results of a computer experiment that was performed using a simple pendulum model as a demonstrative example. We simulated (see figure 5) a pendulum experiment in the realm of deterministic computer experiment to mimic the real life pendulum experiment. We were able to conduct the experiment with two input variables and a univariate output at a specified mass of 0.1kg. The pendulum spent the same minimum time to return to rest at different pendulum lengths and displacement angle of 180°. The maximum time of 19s was spent by the pendulum to stop when L=1m at displacement angle of 150°.

This study has taken a novel application area in the realm of deterministic computer experiment different from the stochastic simulation experiment. The study can be improved if a space-filling design is employed to considerably reduce the simulation time and the mass can also be made as an input variable in addition to the two (displacement angles and pendulum lengths) employed here. The mass may be varied as we did for pendulum length and pendulum displacement angle to simulate the stoppage time.

We have the intention to approximate the pendulum model using a metamodel so as to predict the output of the pendulum computer experiment at untried inputs for future research.

## REFERENCES

[B+02] **M. Bayarri, J. O. Berger, D. Higdon, M. Kennedy, A. Kottas, R. Paulo, J. Sacks, J. Cafeo, J. Cavendish, J. Tu -**

*A framework for the validation of computer models*, in Proceedings of the Workshop on Foundations for V&V in the 21st Century, D.Pace and S.Stevenson, eds., Society for Modelling and Simulation International, 2002.

[Mon01]

**D. C. Montgomery** - *Design and Analysis of Experiments*, 5<sup>th</sup> Edition. John Wiley & Sons, Inco., New York, 2001.

[MC03]

**H. J. Motulsky, A. Christopoulos** - *Fitting Models to Biological Data using Linear and Nonlinear Regression*. A Practical Guide to Curve Fitting (2003). GraphPad, Software Inc., San Diego CA, [www.graphpad.com](http://www.graphpad.com).

[Par00]

**J. E. Parks** - *The Simple Pendulum*. Department of Physics and Astronomy, The University of Tennessee Knoxville, Tennessee 37996-1200, 2000.

[Str03]

**S. Strogatz** - *The Real Scientific Hero of 1953*, New York Times. March 4, Editorial/Op-Ed, 2003.

[SSW89]

**J. Sacks, S. B. Schiller, W. J. Welch** - *Designs for Computer Experiments*, Technometrics 31, 41–47, 1989.