ME222 : NATURE AND PROPERTIES OF MATERIALS

GROUP PROJECT REPORT

GROUP 2

Group Members

- Parul Singh (200674)
- Juee Chandrachud (200480)
- Atharv Dusane (200356)
- Aayushi Mishra (200009)

Acknowledgement

We sincerely thank our course instructors Prof. C. Chandraprakash and Prof. Manjesh Singh for giving us the opportunity to do this project, which helped us gain practical knowledge and hands-on experience in the course ME222. We are grateful for their guidance throughout the project.

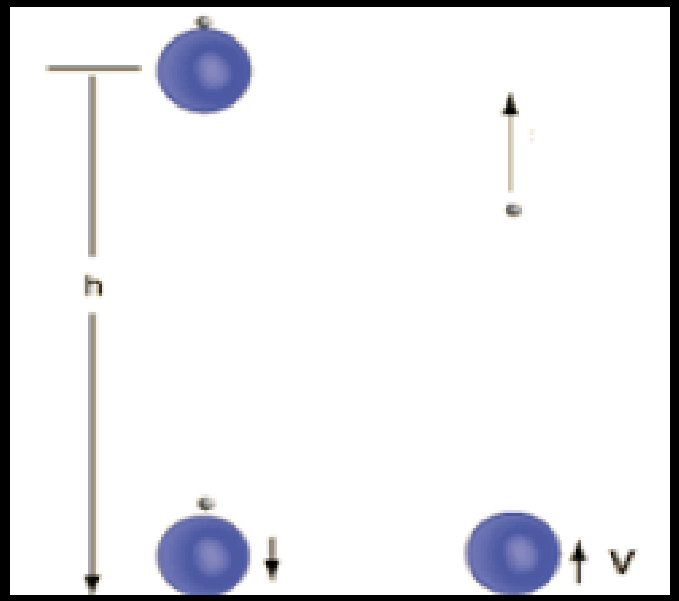
CALCULATING COEFFICIENT OF RESTITUTION OF DIFFERENT SPHERICAL BODIES HITTING A FLAT SURFACE

Contents

Introduction

- Coefficient of restitution(e) is the ratio of velocities of two entities after and before collision.
- It depends on number of factors including material of the colliding bodies.
- Typically, e lies between 0 to 1

$$\text{Coefficient of restitution } (e) = \frac{|\text{Relative velocity after collision}|}{|\text{Relative velocity before collision}|}$$



Formula

- The general formula for coefficient of restitution is - $e = \frac{v_{fb} - v_{fa}}{v_{ia} - v_{ib}}$

  Where,

  v_{fa} - final velocity of A

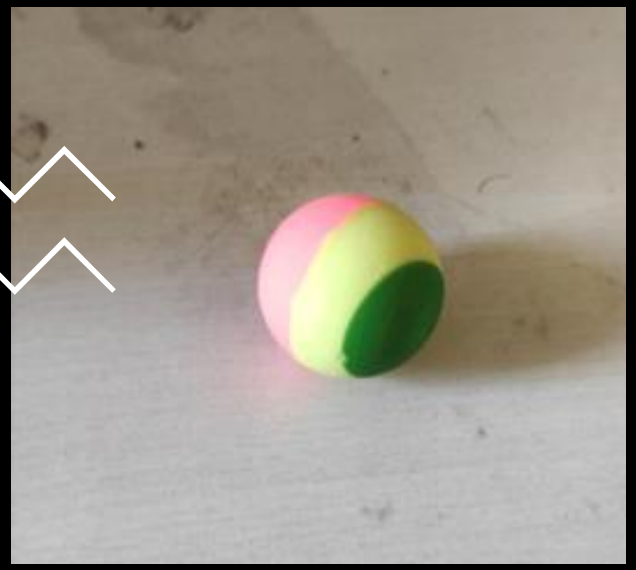
  v_{fb} – final velocity of B

  v_{ib} – initial velocity of A

  v_{ib} – initial velocity of B

- In our experiment, the velocity of body B (ground) is zero and hence the formula is reduced to -

  $e = |v_f| / |v_i|$

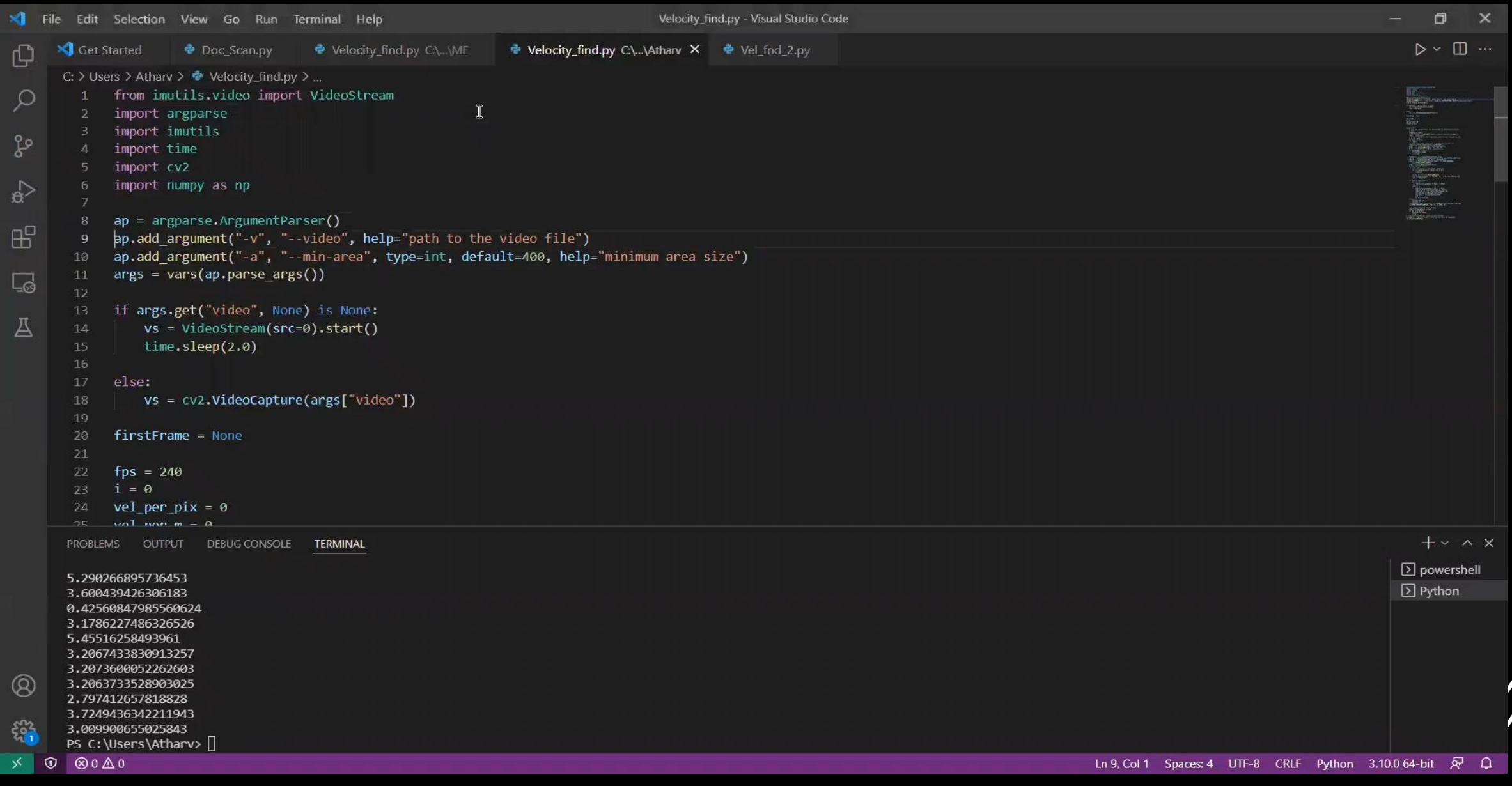


Materials Required

- Spherical Balls – Rubber ball and Table Tennis ball
- Smooth hard surface floor
- Ruler
- Smartphone
- Clamp and pins
- Ruler

Software - OpenCV

- Used in video processing.
- Is integrated with various libraries for the analysis of the performed experiment.
- Identifies the ball and calculates its velocity.
- Our code is displayed in the next slide.


File Edit Selection View Go Run Terminal Help
Velocity_find.py - Visual Studio Code
Get Started
Doc_Scan.py
Velocity_find.py C:\...\ME
Velocity_find.py C:\...\Atharv
Vel_fnd_2.py
C: > Users > Atharv > Velocity_find.py > ...
from imutils.video import VideoStream
import argparse
import imutils
import time
import cv2
import numpy as np
ap = argparse.ArgumentParser()
ap.add_argument("-v", "--video", help="path to the video file")
ap.add_argument("-a", "--min-area", type=int, default=400, help="minimum area size")
args = vars(ap.parse_args())
if args.get("video", None) is None:
vs = VideoStream(src=0).start()
time.sleep(2.0)
else:
vs = cv2.VideoCapture(args["video"])
firstFrame = None
fps = 240
i = 0
vel_per_pix = 0
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
5.290266895736453
3.600439426306183
0.42560847985560624
3.1786227486326526
5.45516258493961
3.2067433830913257
3.2073600052262603
3.2063733528903025
2.797412657818828
3.7249436342211943
3.009900655025843
PS C:\Users\Atharv>
powershell
Python
Ln 9, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.0 64-bit

Experimental Setup

- A clamp is set at the required heights to drop the ball.
- A scale is set at the back to use it for giving pixel to height ratio in OpenCV.
- A smartphone is set using pins and clamps to record the videos.

Procedure

- Drop the ball from certain specific heights from the clamp to ensure that the ball falls vertically.
- Record the video using the smartphone.
- Instantaneous velocity of the ball before and after collision is measured using OpenCV.
- Appropriate calculations are performed to find the coefficient of restitution.

File Edit Selection View Go Run Terminal Help

Velocity_find.py - Visual Studio Code

Get Started | Doc_Scan.py | Velocity_find.py C:\...\ME | Velocity_find.py C:\...\Atharv | Vel_fnd_2.py

C: > Users > Atharv > Velocity_find.py > ...

```
73                  i = 0
74                  print(vel_per_m)
75          else :
76              vel_per_pix = 0
77              vel_per_m = 0
78          cv2.putText(frame, "Velocity: {}".format(str(vel_per_m)), (10, 20),
79          cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
80
81          cv2.imshow("Security Feed", frame)
82          key = cv2.waitKey(1) & 0xFF
83          if key == ord("q"):
84              print(frame.shape)
85              break
86      # cleanup the camera and close any open windows
87      vs.stop() if args.get("video", None) is None else vs.release()
88      cv2.destroyAllWindows()
89
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
2.5032830006313707
2.589944497474801
2.649130163846239
2.5412303245672163
2.792743107143942
2.3345444215628026
2.3350526124961295
2.1953719965247345
2.082959601894621
1.8862630466745618
1.655318134075743
PS C:\Users\Atharv> python velocity_find.py --video Drop_5.mp4
```

powershell
Python

Ln 89, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.0 64-bit

108°F Sunny

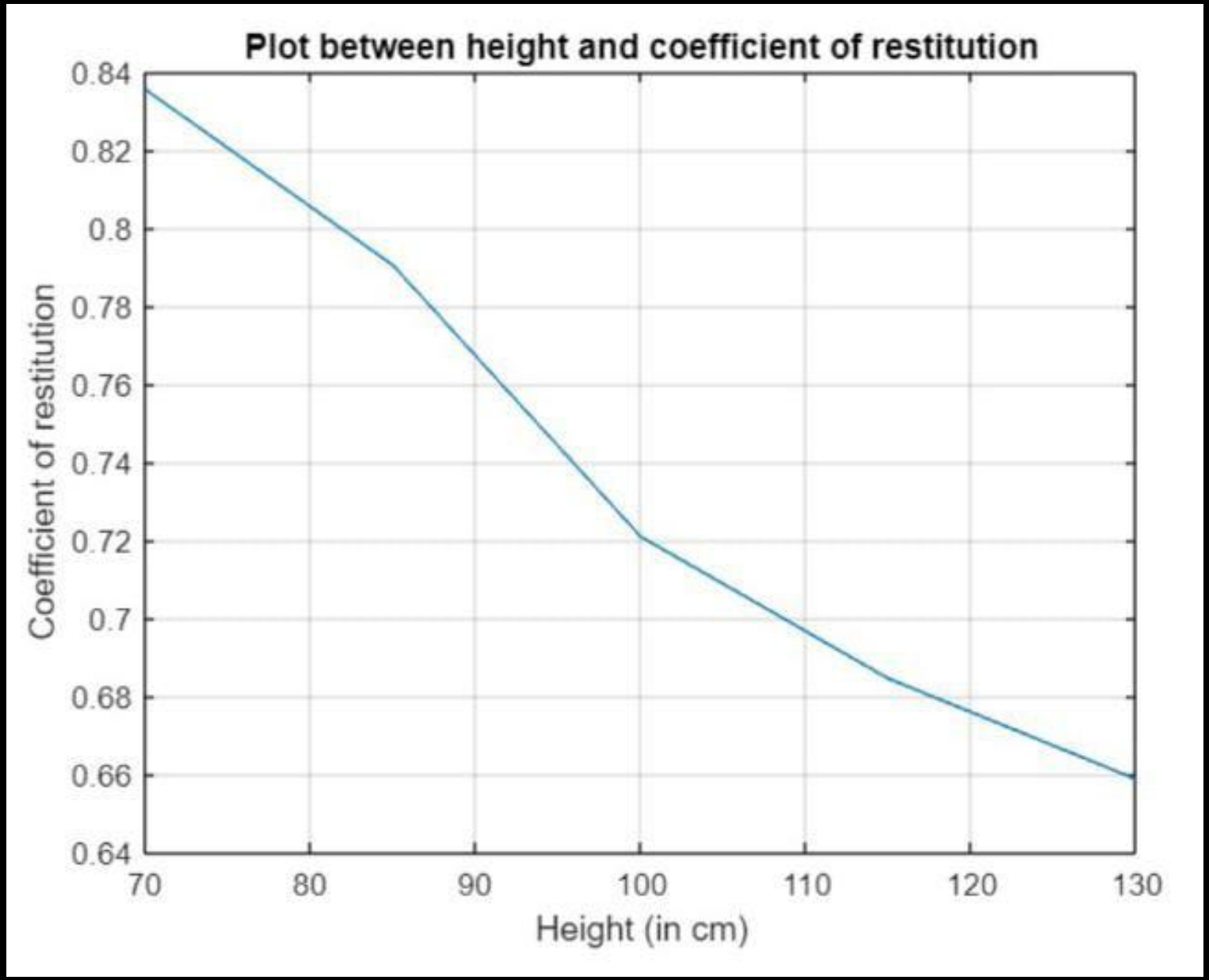
15:35 08/04/2022

Observations for Rubber ball

Observation Table

Sr. No	Height of ball drop - h (m)	Initial velocity – Vi (m/s)	Final velocity – V_f (m/s)	Coefficient of restitution – $e = V_f/V_i$
1	0.70	3.10793	2.56539	0.82543
2	0.85	3.45045	2.72716	0.79037
3	1.00	3.88551	2.80101	0.72088
4	1.15	4.54763	3.11294	0.68452
5	1.30	5.29351	3.48709	0.65875

Graph

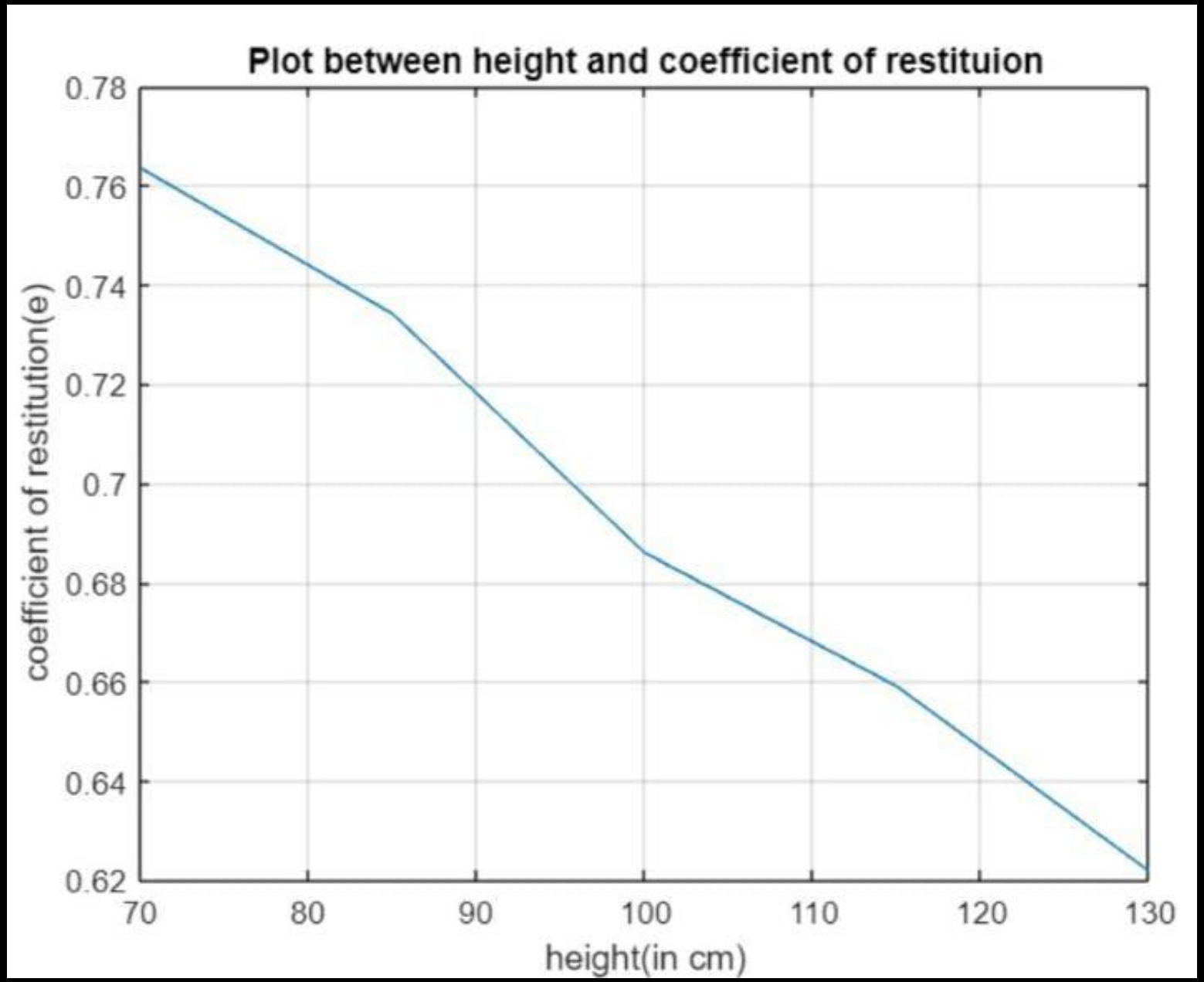

Plot between height and coefficient of restitution
Coefficient of restitution
0.84
0.82
0.8
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0.64
70
80
90
100
110
120
130
Height (in cm)

Observations for Table Tennis ball

Observation Table

Sr. No	Height of ball drop - h (m)	Initial velocity – Vi (m/s)	Final velocity – V_f (m/s)	Coefficient of restitution – $e = V_f/V_i$
1	0.70	2.94515	2.23871	0.76353
2	0.85	3.31237	2.43174	0.73414
3	1.00	3.65479	2.50711	0.68598
4	1.15	4.46898	2.94537	0.65907
5	1.30	5.17342	3.21703	0.62184

Graph


Plot between height and coefficient of restituion
coefficient of restitution(e)
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0.64
0.62
70
80
90
100
110
120
130
height(in cm)

Experiment Analysis

- From the above observations, it can be seen that the coefficient of restitution is different for different materials – rubber and plastic and thus it is a material property.

- The coefficient of restitution is observed to be inversely proportional to the height from which the ball is dropped

Sources of Error

- The surface is not perfectly flat.
- The camera angle also causes some errors in the readings.
- Errors while detecting the ball in OpenCV affects the velocity obtained.
- Error while measuring the height from which the ball is dropped.

Conclusion

- We have successfully analysed the variation of coefficient of restitution with change in the material of the colliding bodies.
- We have also observed the trend of change in the coefficient of restitution with a change in height from which the ball is dropped.