SUMMER OF SOFTWARE

Content

- Fusion 360
- Matlab
- Simulink
- Ansys

Why CAD-Modelling!!

Computer-aided design (**CAD**) is the use of computer (or workstation) to aid in the creation, modification, analysis, or optimization of a design.

1. Used to increase the productivity of the designer.
2. Improve the quality of design.
3. Improve communications through documentation.
4. Create a database for manufacturing.

Majorly used software For cad

SolidWorks

AUTODESK® FUSION 360™

But we have used Fusion 360 because

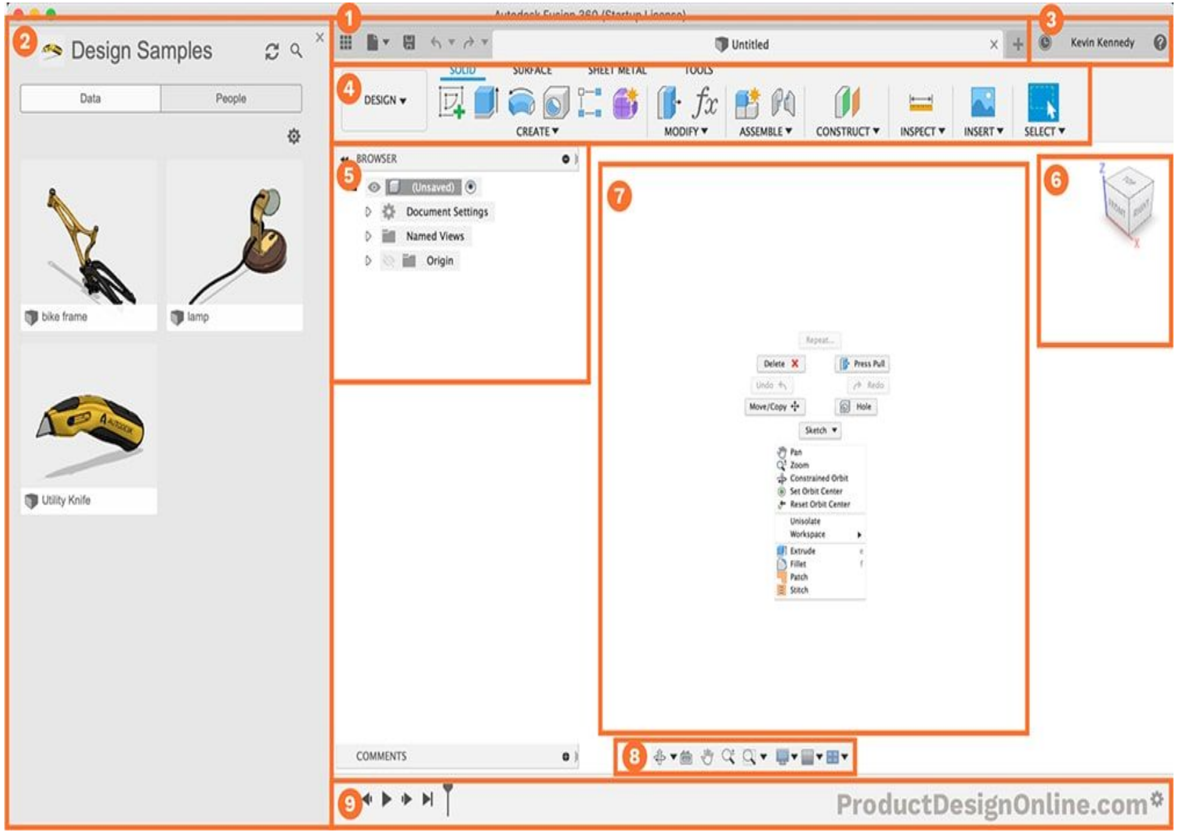
- Cloud based software so light weighted
- Educational licence is available easily
- Easy to use
- Can access software at any platform (even android)

AUTODESK®

FUSION 360™

Basic functionality of Fusion 360

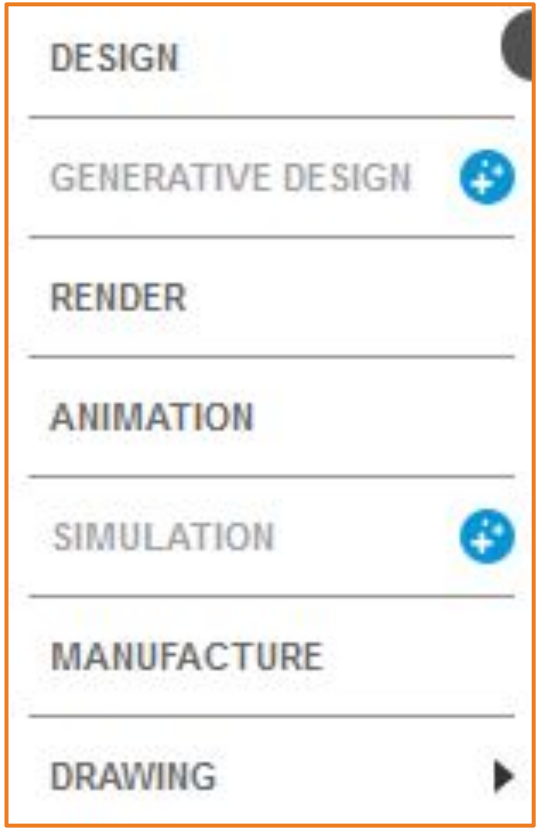
- Application Bar
- Data Panel
- Profile and Help
- Toolbar
- Browser
- ViewCube
- Canvas and Marking Menu
- Navigation Bar and Display Settings
- Timeline



Workspaces

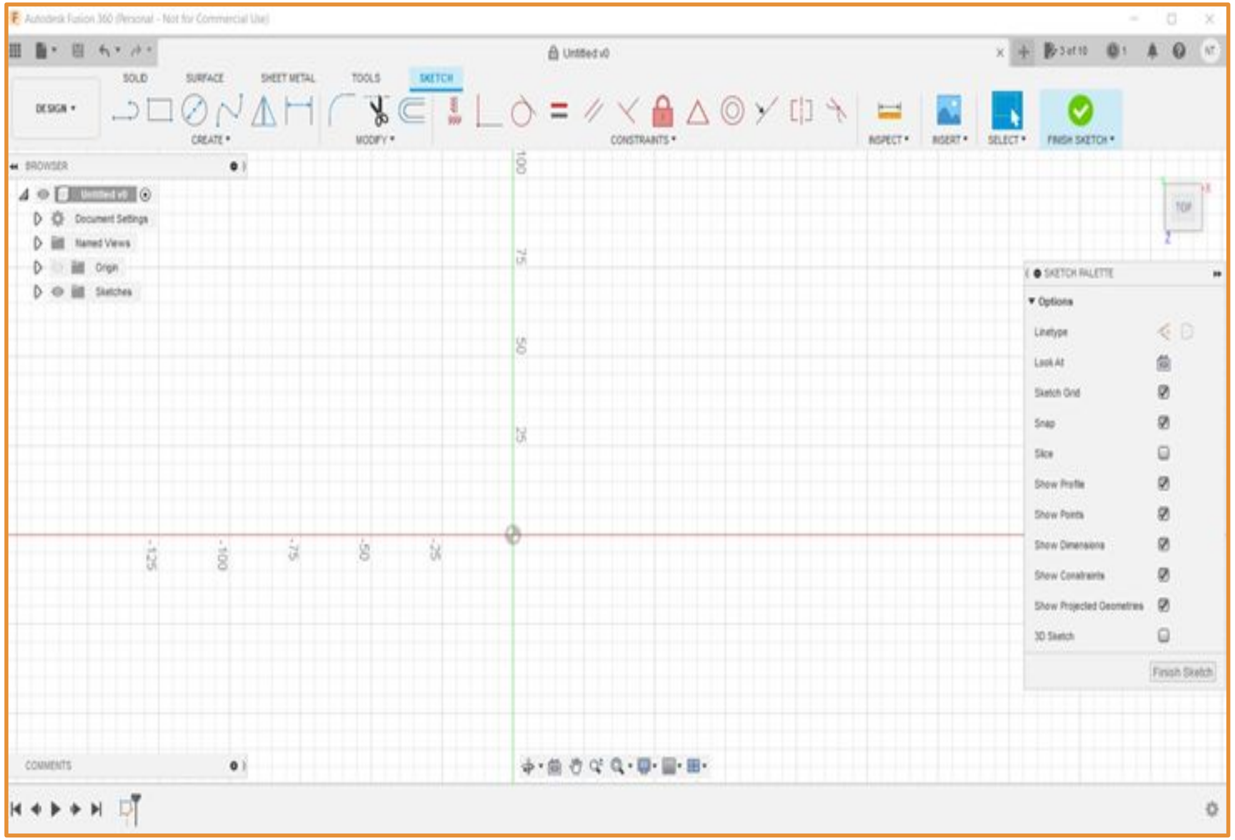
Main Focus :-

- Design
- Animation
- Simulation
- Drawing
- Generative design



2D sketching

- Line
- Rectangle
- Circle
- Dimension
- Fillet
- Trim
- Equal
- Parallel
- Perpendicular
- Symmetry



SKETCH PALETTE

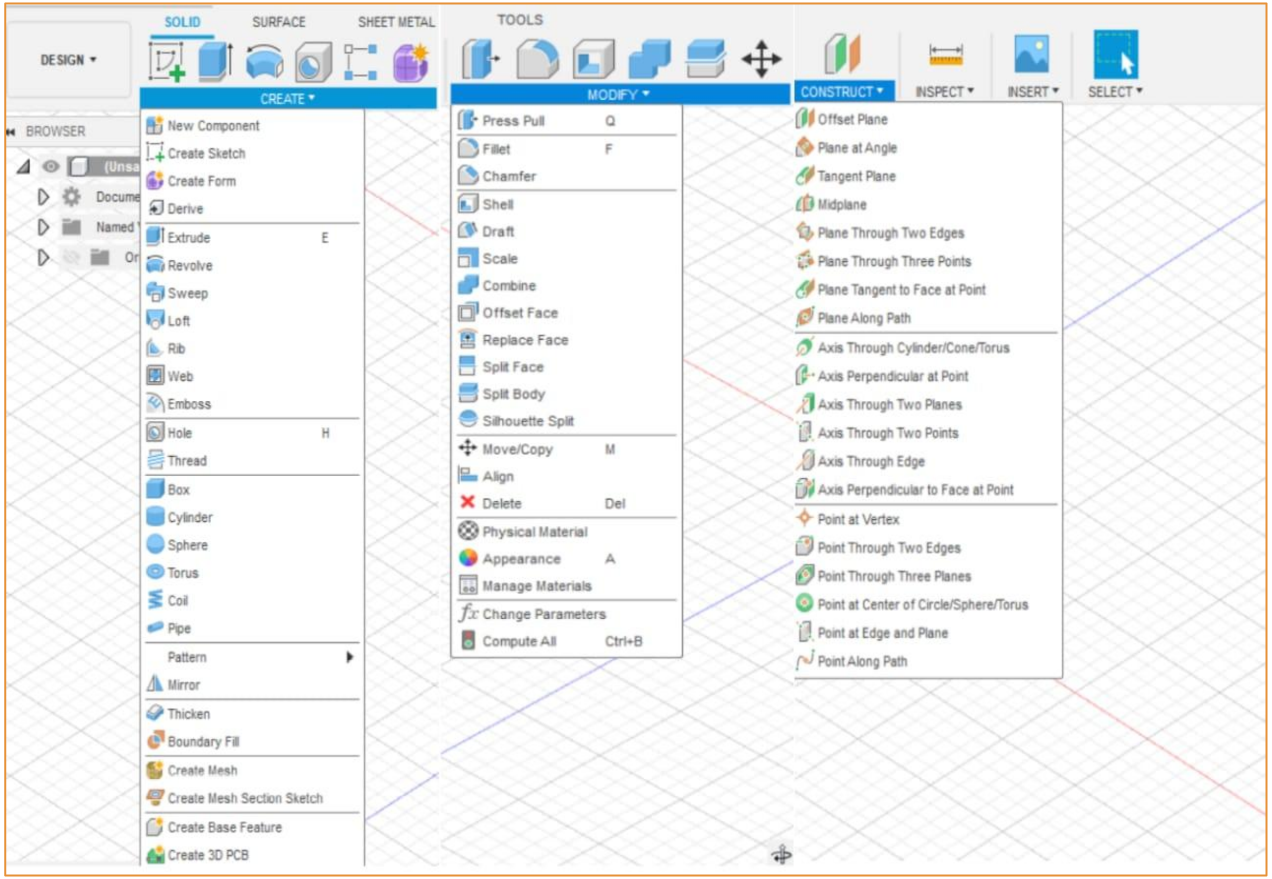
Options

Linetype	
Look At	
Sketch Grid	✓
Snap	✓
Slice	☐
Show Profile	✓
Show Points	✓
Show Dimensions	✓
Show Constraints	✓
Show Projected Geometries	✓
3D Sketch	☐

Finish Sketch

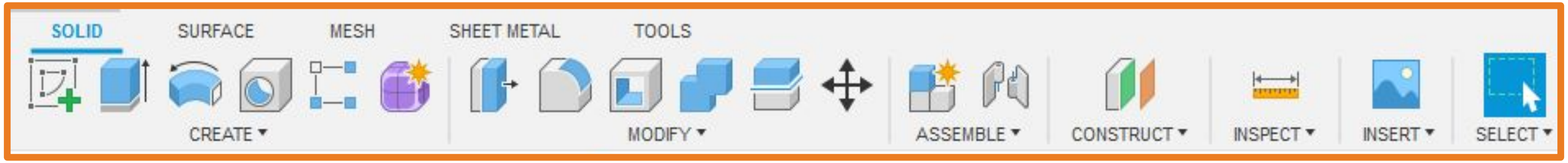
3D Modelling

- Extrude
- Revolve
- Hole
- Fillet
- Chamfer
- Sweep
- Loft
- Construct
- Project

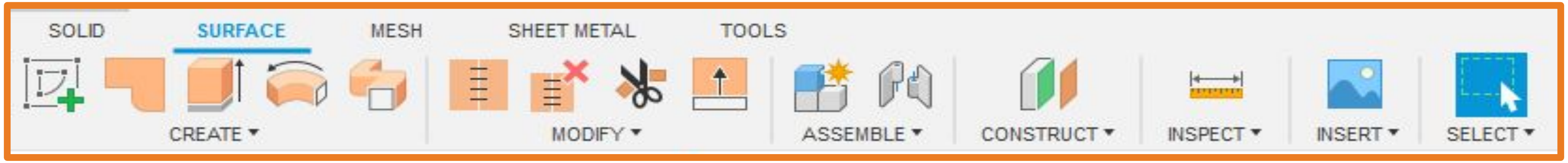


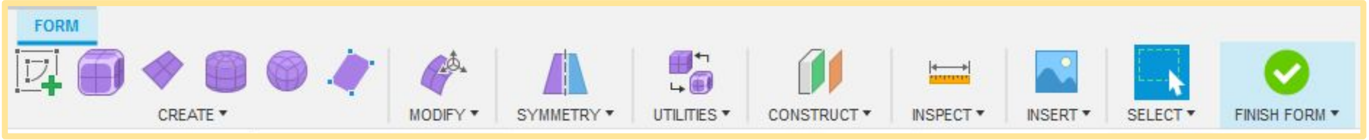
Design Workspace

Solid Modelling

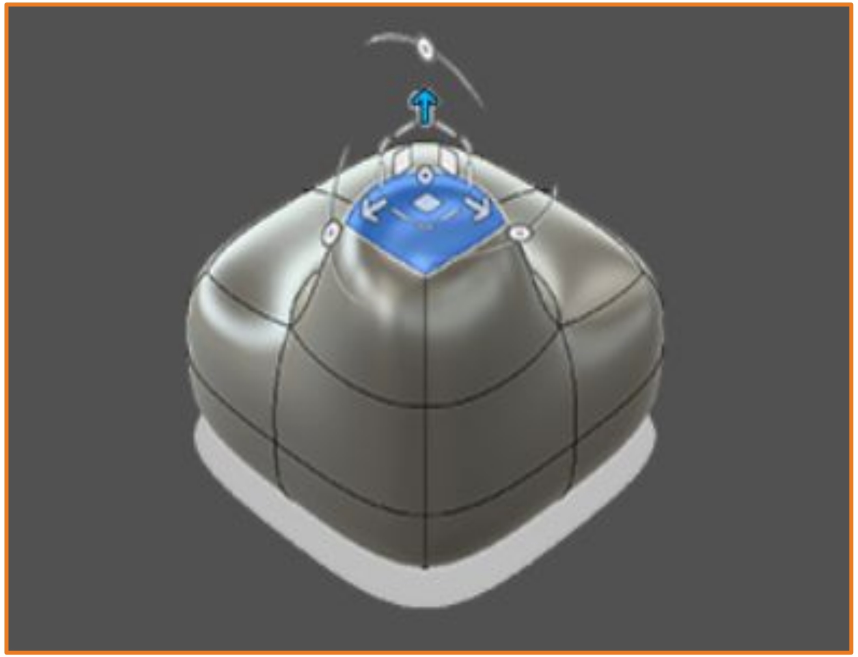


Surface Modelling





Freeform modelling is a technique used in cad to describe or generate the skin of a 3D geometric element.

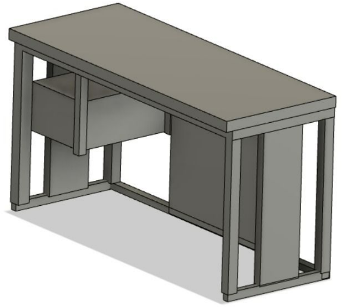


FreeForm Modelling

Assignment 1

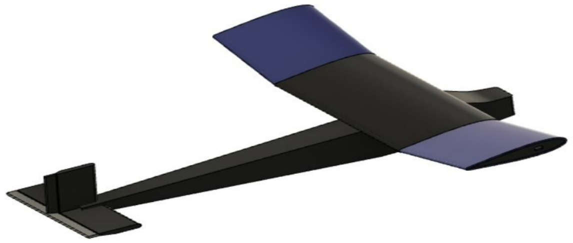
Assignment 2

Grinder



Study table

Headphone



Aeroplane

Fan

Mobile phone

MATLAB

MATLAB®

# Use of Matlab:-

1. Math and computing.
2. Algorithm development.
3. Modelling,simulation.
4. Data analysis,exploration.
5. Scientific and engineering graphics.

So all of us are well aware of the fact what we can achieve with matlab
But it is too vast so what we have done ?

We develop into the world of MATLAB through really simple yet proficient courses listed below:

1. MATLAB Basics
2. Solving Nonlinear Equations with MATLAB
3. Solving Ordinary Differential Equations with MATLAB

MATLAB®

MATLAB Basics

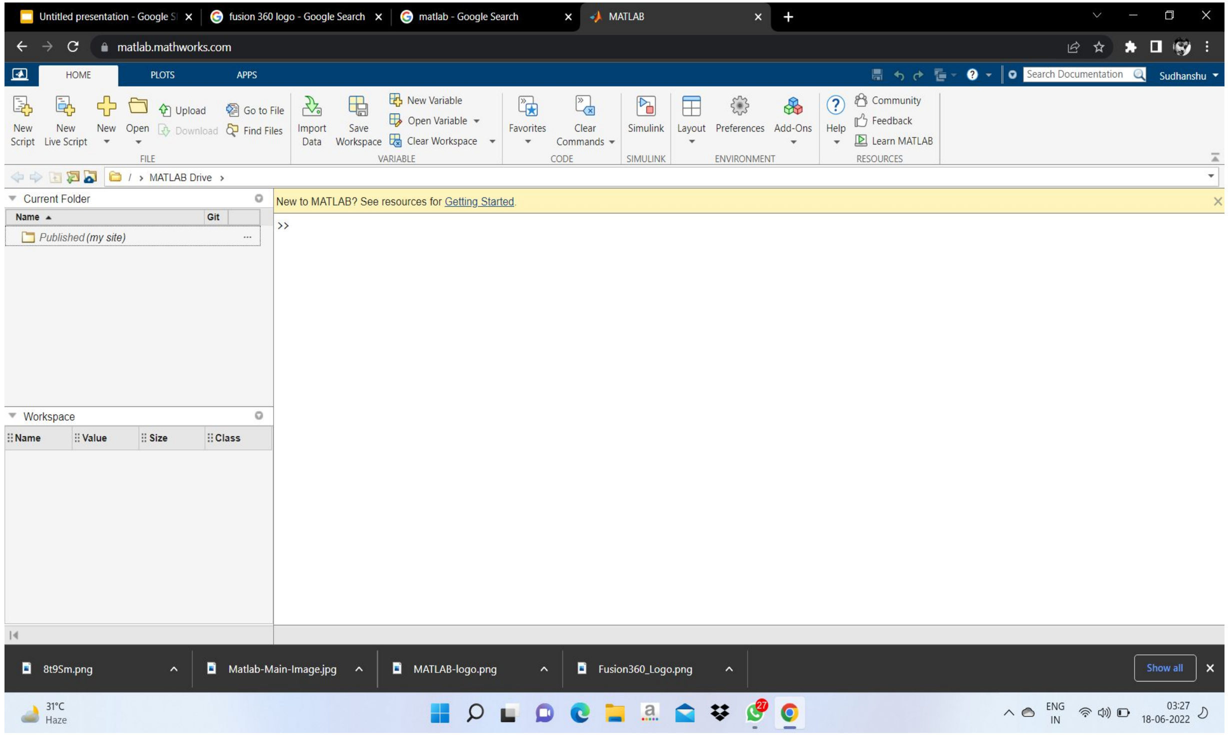
The tutorials include:

1. Commands
2. MATLAB desktop and editor
3. Vectors and Matrices
4. Indexing and Modifying Arrays
5. Array Calculations
6. Calling functions
7. Plotting Data
8. Importing Data
9. Logical Arrays
10. Programming

MATLAB®

#Matlab interface

1. File.
2. Variable.
3. Code.
4. Simulink.
5. Environment.
6. Resources.
7. Workspace.
8. Current folder.



New to MATLAB? See resources for Getting Started.

```
>>
```

Solving nonlinear equation with matlab

The course aims to let learners get familiar with the basics to solving nonlinear equations in MATLAB.

1. Root Finding
2. Finding a Root: The Bisection Method
3. Finding a Root: The fzero Function
4. Systems of Nonlinear Equations: The fsolve Function

Solving Ordinary Differential Equations with MATLAB

This course aimed at learning the basics of solving ordinary differential equation in matlab.

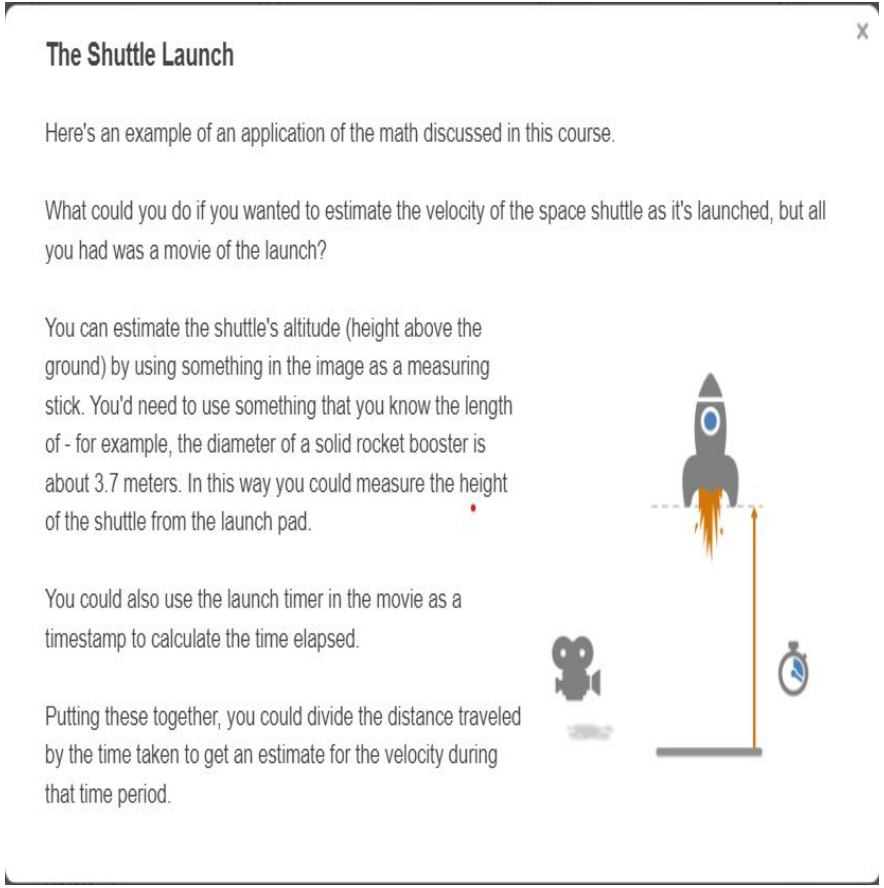
1.What is an Ordinary Differential Equation?

Differential equations were introduced in this module, and what it means to find a solution to a differential equation.

2. Solving ODEs Numerically

3. Solving Systems of ODEs Numerically

PROJECT- ESTIMATING SPACE SHUTTLE VELOCITY



The Shuttle Launch

Here's an example of an application of the math discussed in this course.

What could you do if you wanted to estimate the velocity of the space shuttle as it's launched, but all you had was a movie of the launch?

You can estimate the shuttle's altitude (height above the ground) by using something in the image as a measuring stick. You'd need to use something that you know the length of - for example, the diameter of a solid rocket booster is about 3.7 meters. In this way you could measure the height of the shuttle from the launch pad.

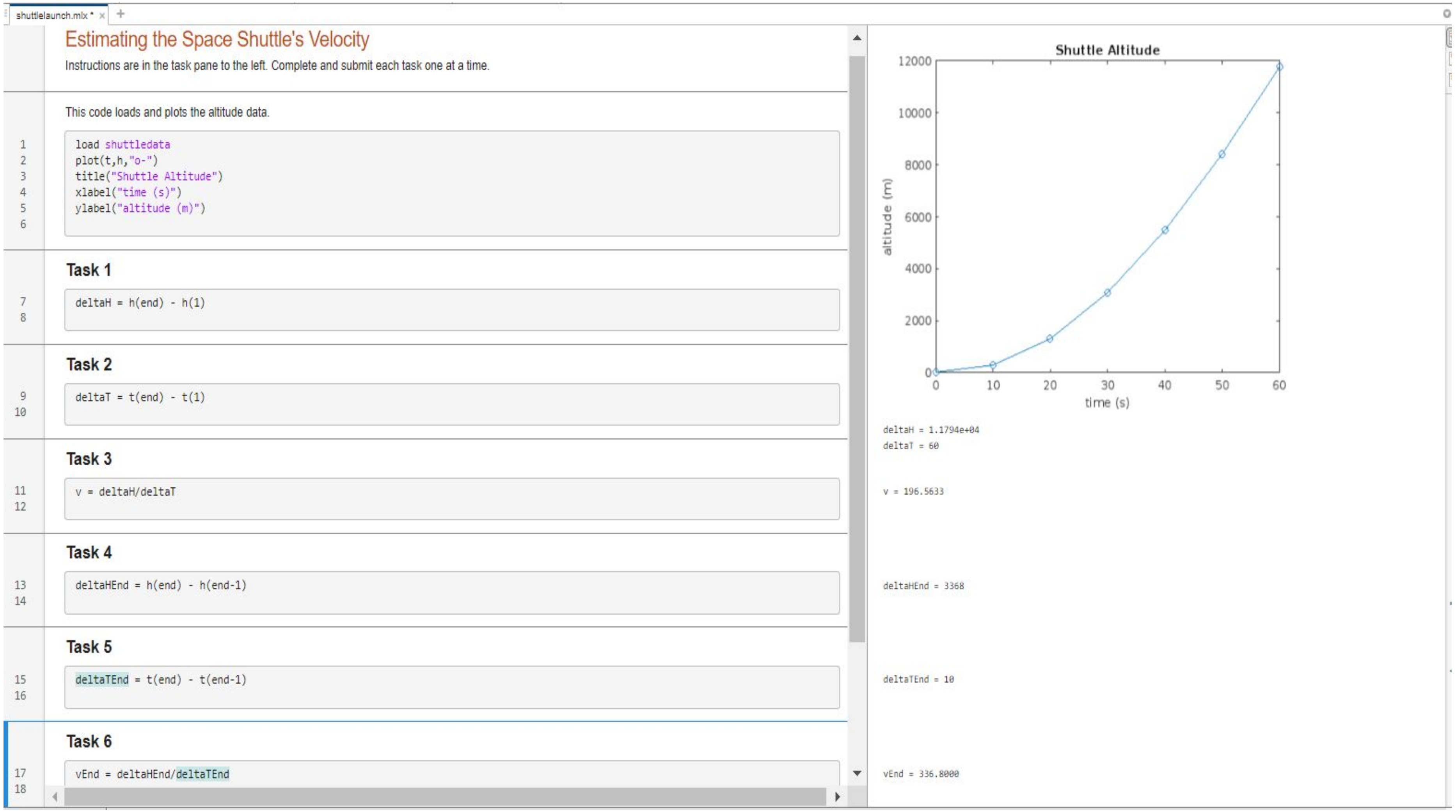
You could also use the launch timer in the movie as a timestamp to calculate the time elapsed.

Putting these together, you could divide the distance traveled by the time taken to get an estimate for the velocity during that time period.

TASK 1- The vector `h` contains the shuttle's altitude data. Find the total change in height by subtracting the first element of `h` from the last element of `h`. Assign the result to the variable `deltaH`.

TASK 2- The vector `t` contains the time (in seconds) for the shuttle altitude measurements. Find the total change in time by subtracting the first and last elements of `t`. Assign the result to the variable `deltaT`.

TASK 3-Find the shuttle's velocity by calculating the change in height divided by the change in time. Assign the result to the variable `v`.
Leave off the final semicolon to display the result.
TASK 4-Find the shuttle's change in height between the final two altitude measurements by taking the difference between the final two elements of `h`. Assign the result to `deltaHEnd`.
TASK 5-Find the change in time between the final two altitude measurements by taking the difference between the final two elements of `t`. Assign the result to `deltaTEnd`.
TASK 6-Calculate velocity from `deltaHEnd` and `deltaTEnd`, and assign the result to the variable `vEnd`.Leave off the final semicolon to display the result.



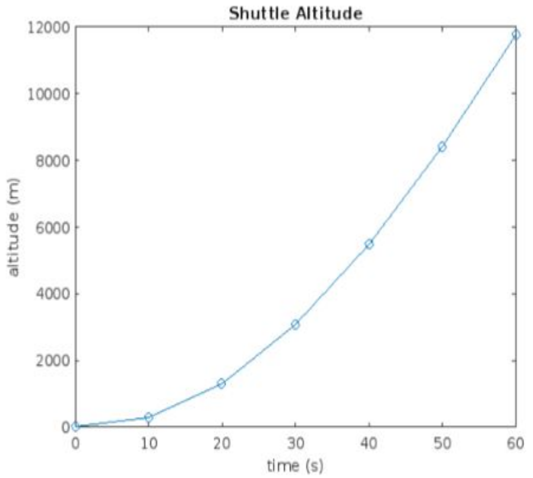
shuttlelaunch.mlx *

Estimating the Space Shuttle's Velocity

Instructions are in the task pane to the left. Complete and submit each task one at a time.

This code loads and plots the altitude data.

```
load shuttledata
plot(t,h,"o-")
title("Shuttle Altitude")
xlabel("time (s)")
ylabel("altitude (m)")
```



Task 1

```
deltaH = h(end) - h(1)
```

```
deltaH = 1.1794e+04
```

Task 2

```
deltaT = t(end) - t(1)
```

```
deltaT = 60
```

Task 3

```
v = deltaH/deltaT
```

```
v = 196.5633
```

Task 4

```
deltaHEnd = h(end) - h(end-1)
```

```
deltaHEnd = 3368
```

Task 5

```
deltaTEnd = t(end) - t(end-1)
```

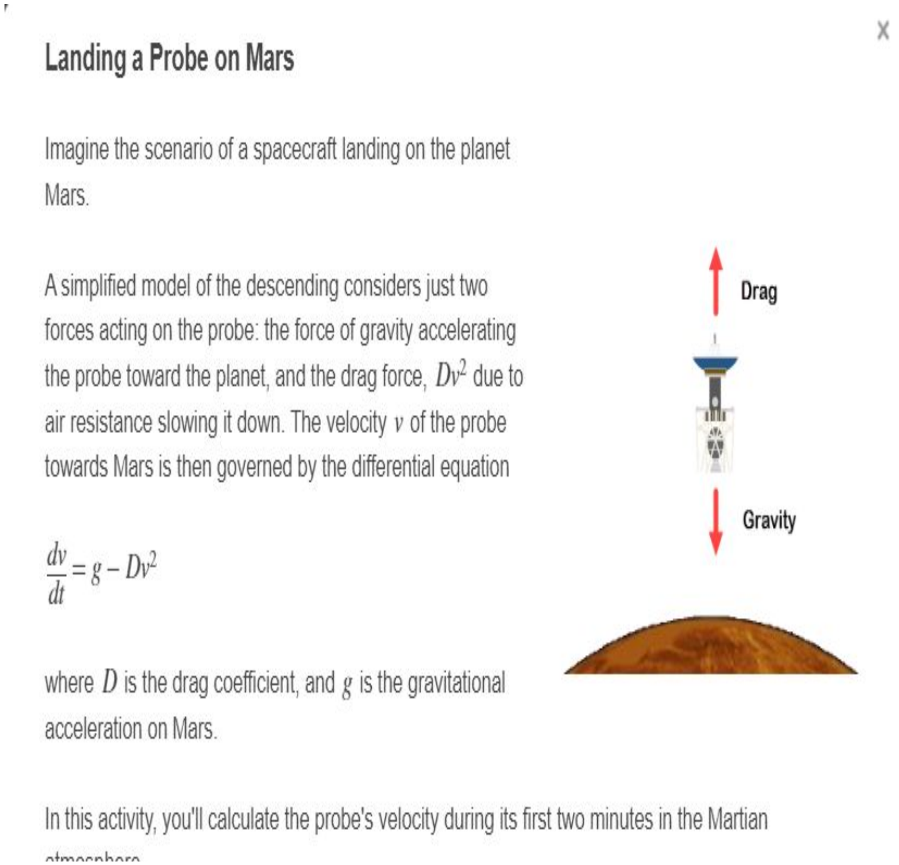
```
deltaTEnd = 10
```

Task 6

```
vEnd = deltaHEnd/deltaTEnd
```

```
vEnd = 336.8000
```

PROJECT- LANDING A PROBE ON MARS



Landing a Probe on Mars

Imagine the scenario of a spacecraft landing on the planet Mars.

A simplified model of the descending considers just two forces acting on the probe: the force of gravity accelerating the probe toward the planet, and the drag force, Dv^2 due to air resistance slowing it down. The velocity v of the probe towards Mars is then governed by the differential equation

$$\frac{dv}{dt} = g - Dv^2$$

where D is the drag coefficient, and g is the gravitational acceleration on Mars.

In this activity, you'll calculate the probe's velocity during its first two minutes in the Martian

TASK 1-Find and plot the probe's velocity from t=0 to 120 seconds.
Use `ode45` to solve the ODE. Define the time interval and initial condition, and create a local ODE function `fallingbody` at the bottom of the script to implement the ODE.
Assign the outputs to the variables `tSol` and `vSol`. Then plot `vSol` as a function of `tSol`.

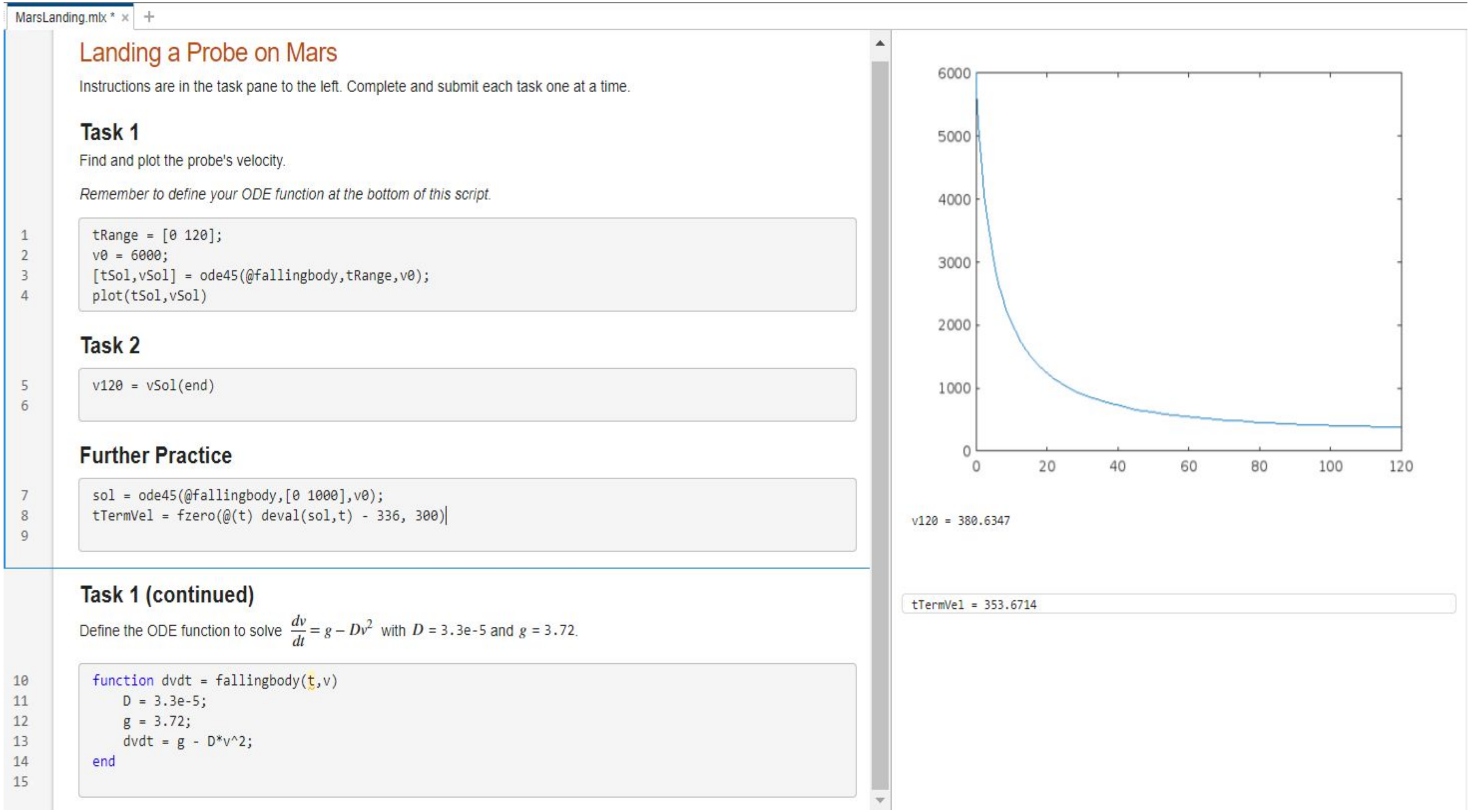
Recall that the number 3.3×10^{-5} can be written in MATLAB as `3.3e-5`.
TASK 2-Find the probe's velocity after 120 seconds. Assign it to the variable `v120`.You may want to leave off the semicolon to display the result
In just two minutes, the probe has slowed from 6000 m/s to 380 m/s.
The probe's *terminal velocity* is reached when the gravitational and drag forces are balanced, so the drag force cannot slow it down further. Here, the terminal velocity is $\sqrt{g/D}\approx$ 336 m/s. About how long does it take for the probe to slow down to 336 m/s?
Since the velocity is only 380 m/s after 120 seconds, you'll need to solve the ODE on a larger time interval.

```
sol = ode45(@fallingbody,[0 1000],v0);
```

Then you can evaluate `sol` at multiple times until the result is close to 336 m/s.`deval(sol,200)deval(sol,400)deval(sol,350)`

Alternatively, if you are familiar with the `fzero` function, you could use it to find when `sol` evaluates to 336:

```
  sol = ode45(@fallingbody,[0 1000],v0);
tTermVel = fzero(@(t) deval(sol,t) - 336, 300)
```



Landing a Probe on Mars

Instructions are in the task pane to the left. Complete and submit each task one at a time.

Task 1

Find and plot the probe's velocity.

Remember to define your ODE function at the bottom of this script.

```
tRange = [0 120];
v0 = 6000;
[tSol,vSol] = ode45(@fallingbody,tRange,v0);
plot(tSol,vSol)
```

Task 2

```
v120 = vSol(end)
```

```
v120 = 380.6347
```

Further Practice

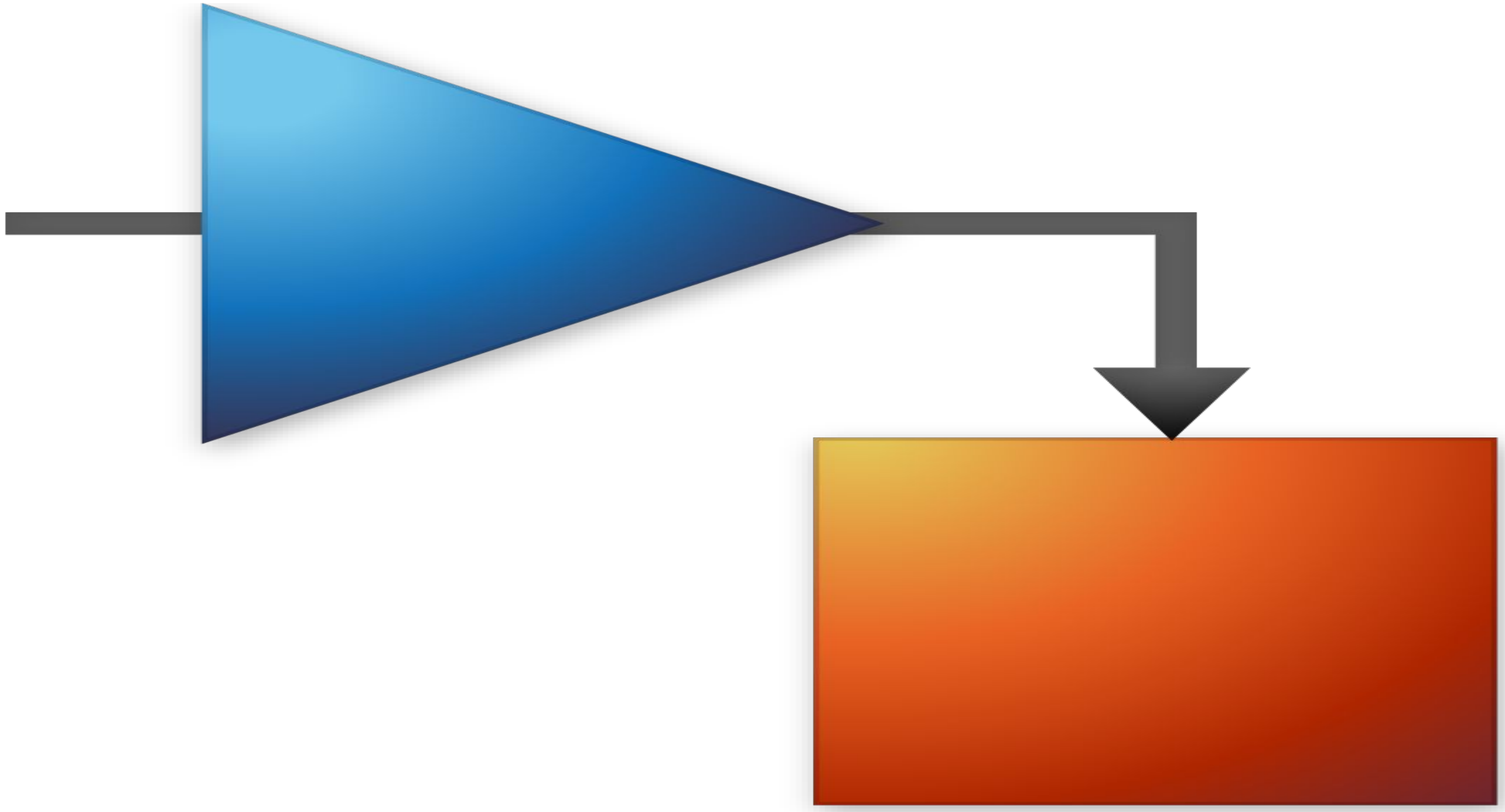
```
sol = ode45(@fallingbody,[0 1000],v0);
tTermVel = fzero(@(t) deval(sol,t) - 336, 300)
```

```
tTermVel = 353.6714
```

Task 1 (continued)

Define the ODE function to solve $\frac{dv}{dt} = g - Dv^2$ with $D = 3.3\text{e-}5$ and $g = 3.72$.

```
function dvdt = fallingbody(t,v)
    D = 3.3e-5;
    g = 3.72;
    dvdt = g - D*v^2;
end
```



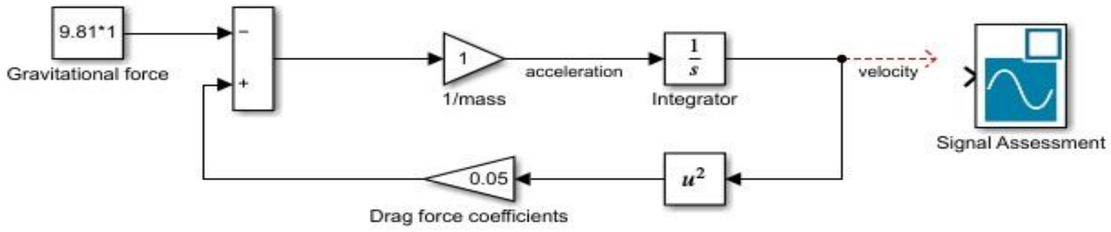
SIMULINK

It is a MATLAB based graphical programming environment for modeling, simulating and analyzing the mechanical, electrical and hydraulic system.

Its interface is a graphical block diagramming tool and a customizable set of block libraries

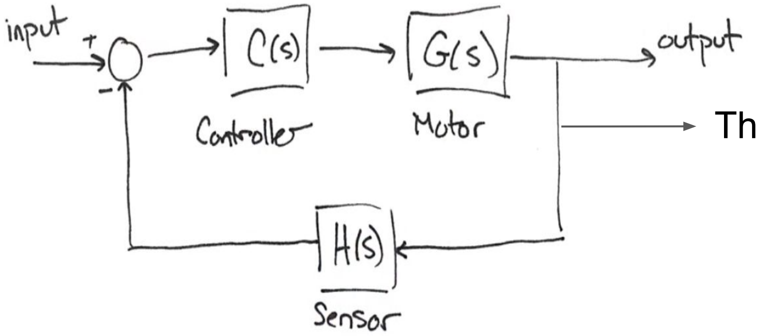
$$\Sigma F = ma$$

$$-mg + \alpha v^2 = ma$$

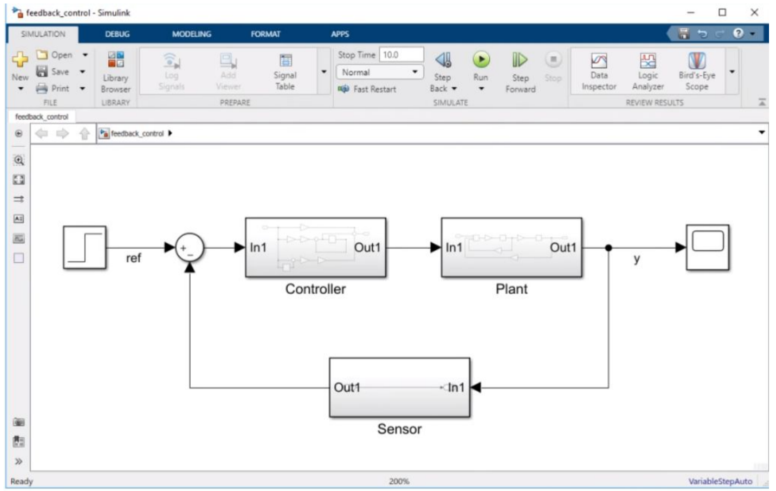


WHY SIMULINK

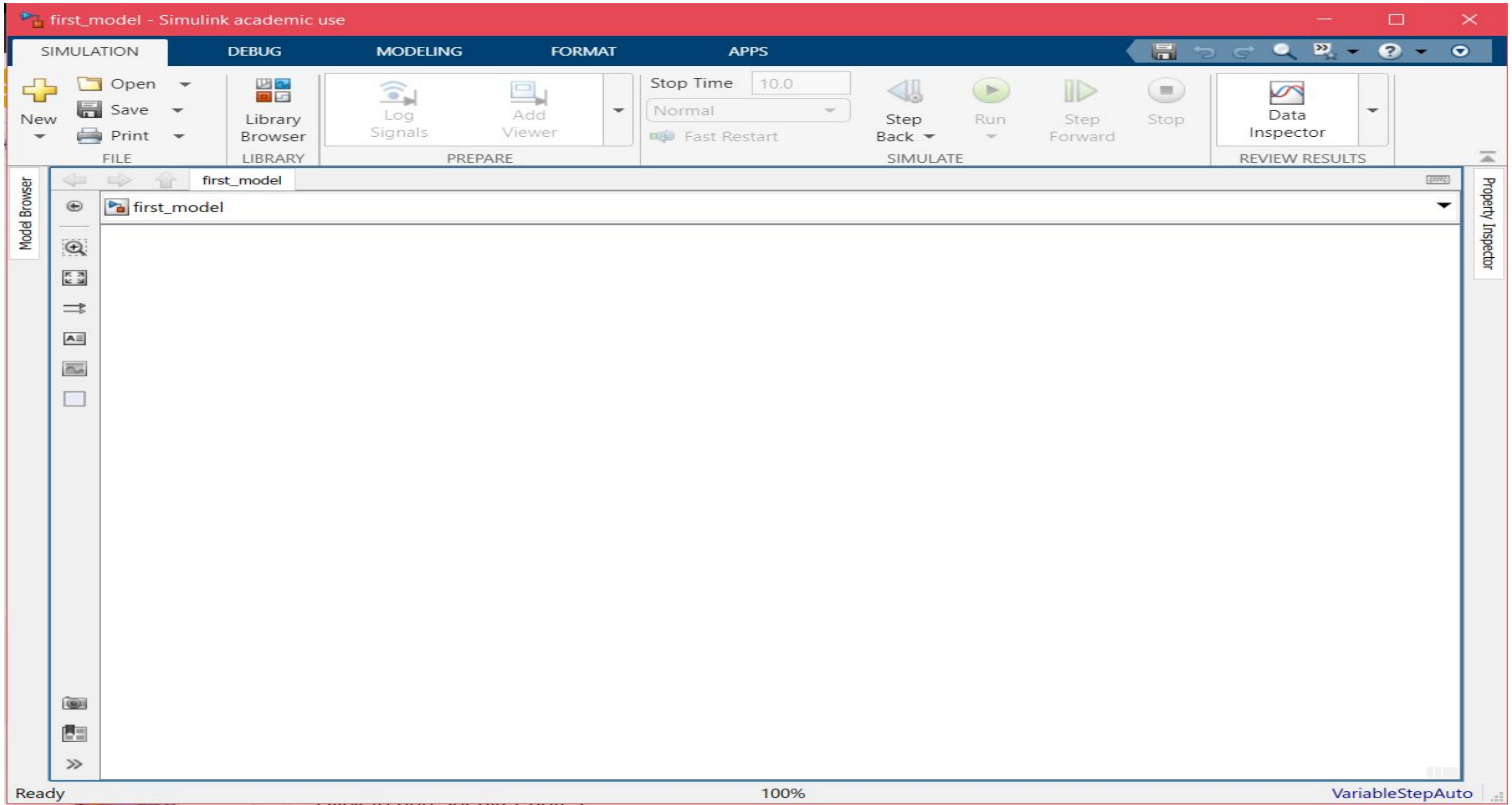
- It is not possible all the time to write the code from scratch to simulate any system, with the pre built libraries and models Simulink does the task.
- Visualization tools for analyzing and comparing results from multiple simulations.
- A library of prebuilt blocks for modeling algorithms and physical systems.



This to This



HOW IT LOOKS OR UI

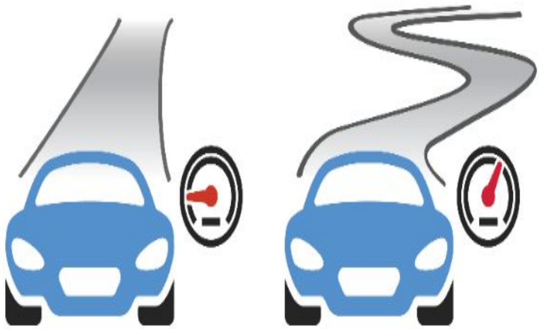


Project - Automotive Performance Package

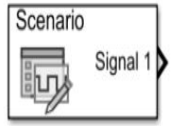
Project - Automotive Performance Package

← PREVIOUS | NEXT →

In this project, you'll simulate the logic for an automotive performance package. Vehicles today have dozens of computers that together adjust the car's behavior in response to data read by hundreds of sensors. In this simplified system, assume that you are responding to two sensors —speed and lateral acceleration—to decide between a "high performance" and "economy" mode.



In order to simulate the sensor data, the model uses a block that you haven't seen yet: the Signal Editor block.



Don't worry, though, Signal Editor is just another type of source, like Sine Wave or Ramp. You can inspect these signals by using what you learned previously, and adding a Scope to the model.

TASK1-Add an Abs (**Simulink > Math Operations**) block to take the absolute value of lateral acceleration.

Add three Compare to Constant blocks to the model. Set their conditions to

≥ 100

≥ 35

≥ 3

Connect the blocks to their appropriate signals

TASK2-Add a Logical Operator block to the model. Use the default **Operator**, `AND`. Connect the appropriate Compare to Constant blocks to create the statement: speed ≥ 35 km/h AND

abs(lat. accel.) ≥ 3 m/s^2

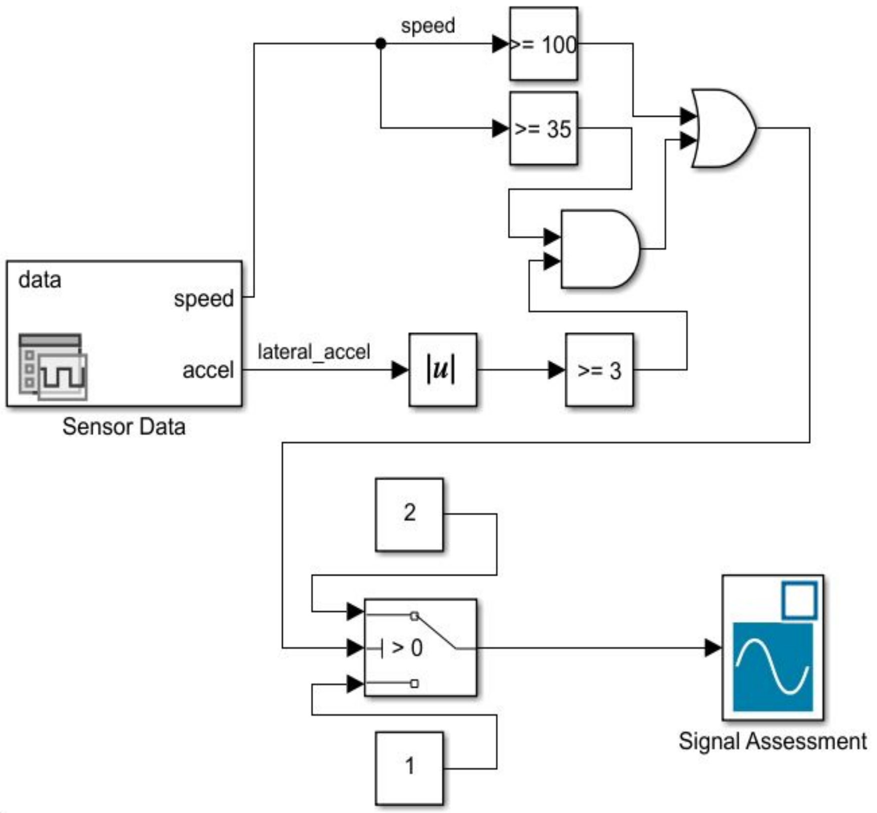
Task3-Connect the output of the AND block to the Signal Assessment block. (You will connect the remaining signal next.)

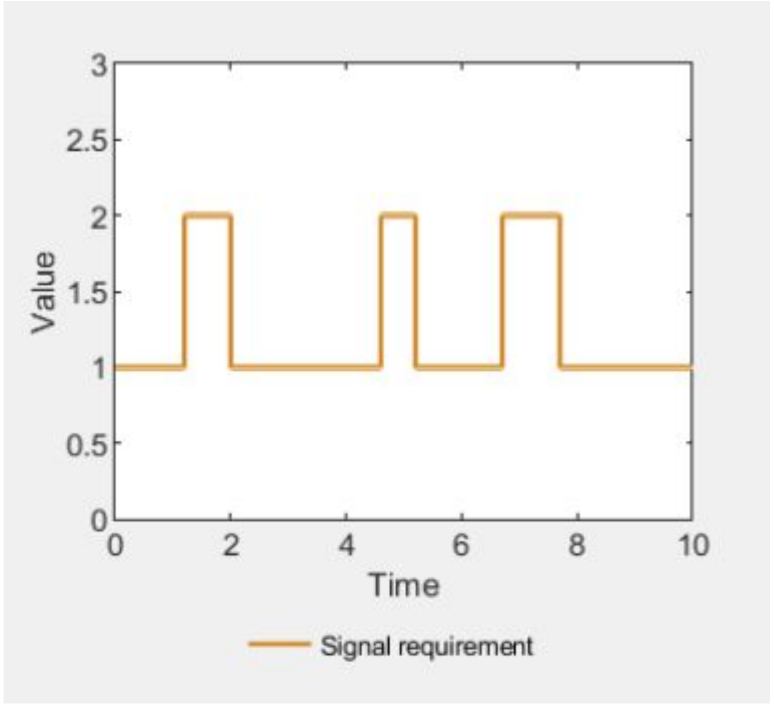
TASK4-Add a second Logical Operator block to the model. Change its **Operator** to `OR`. Connect the blocks to create the statement: (speed ≥ 100 km/h) OR (speed ≥ 35 km/h AND

abs(lat. accel.) ≥ 3 m/s^2)

Connect the output of the OR block to the Signal Assessment block

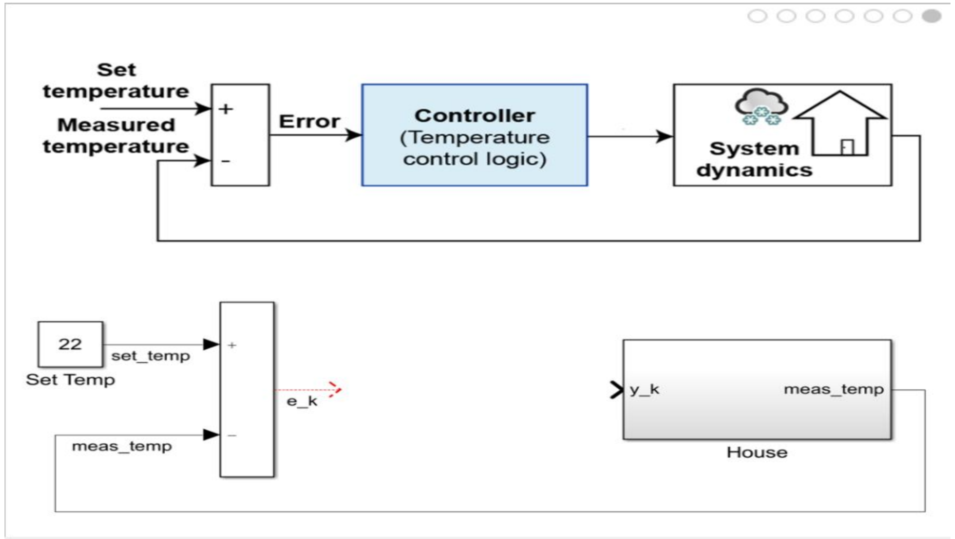
TASK5-Add a Switch block to the model and connect the output from the previous task to the control signal. Add two Constant blocks, with values `1` and `2`, to represent the driving modes. Connect the blocks to the appropriate locations on the Switch block.


speed
>= 100
>= 35
data
speed
accel
Sensor Data
lateral_accel
|u|
>= 3
2
> 0
1
Signal Assessment


Value
Time
Signal requirement
0
0.5
1
1.5
2
2.5
3
0
2
4
6
8
10

Project - Thermostat Model

In Chapter 10, you modeled a first order cooling equation. That equation could represent the temperature dynamics of a house. To keep the house comfortable, you might want to control the temperature, using some strategy based on the current and desired temperatures. This is called a *control system*, and is exactly what a thermostat is! Since most controllers are digital, they are often modeled as discrete systems



In this project, you will model the discrete equations representing PI control. The model will contain a set temperature and a block representing the house, similar to the model you built previously. You will create the controller.

TASK1-Add another Gain block to the model and set its value to `Ki*Ts`. Branch `e_k` and connect it to the Gain block you just created, to create the term $K_iT_se[k]$.

TASK2-Add the appropriate number of Unit Delay blocks to the model to represent the integrator term, $y_i[k-1]$. You do not need to make any connections yet.

TASK3-Create stubs on the Unit Delay block. Label the appropriate signals `yi_k` and `yi_kminus1`.

TASK4-Use an Add block to sum the terms of the integrator equation, $K_iT_se[k]+y_i[k-1]$. Do not make any other connections yet.

TASK5-Make the connections that complete the integrator equation $y_i[k]=K_iT_se[k]+y_i[k-1]$. (Do not connect $y_p[k]$ yet.) Branch `yi_k` to the Signal Assessment block.

TASK6-

1. Add the signals `yp_k` and `yi_k` to create $y[k]$.
2. Connect $y[k]$ to the input port of the House subsystem.
3. Branch and connect the temperature, `meas_temp`, to the Signal Assessment block.

TASK7-Set the **Sample time** to `Ts`. Use the default **Initial condition** of `0`.

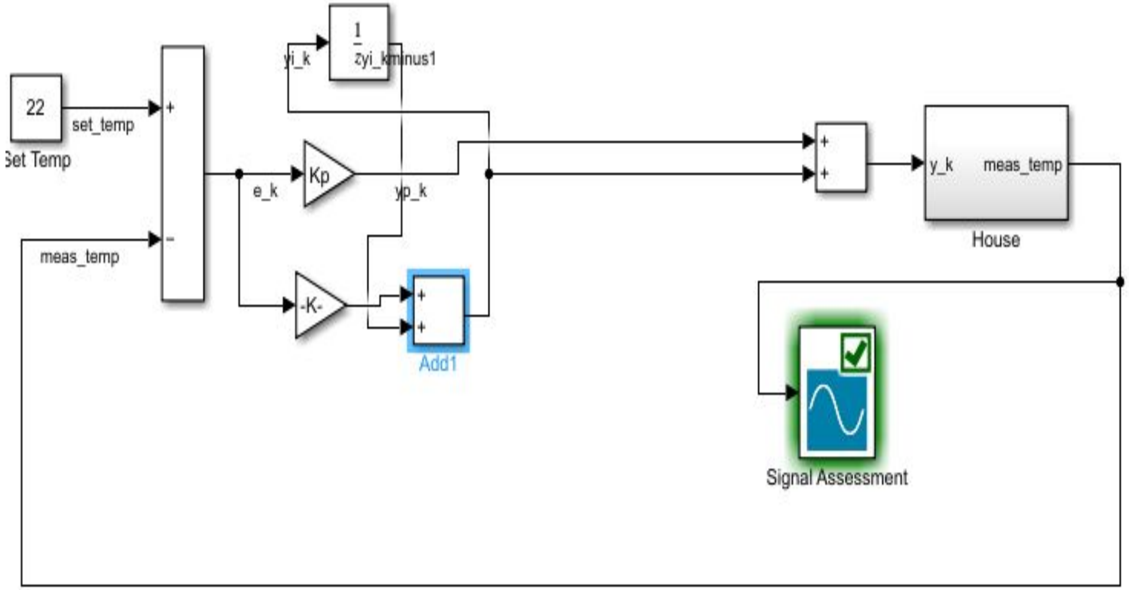
Proportional-integral control:

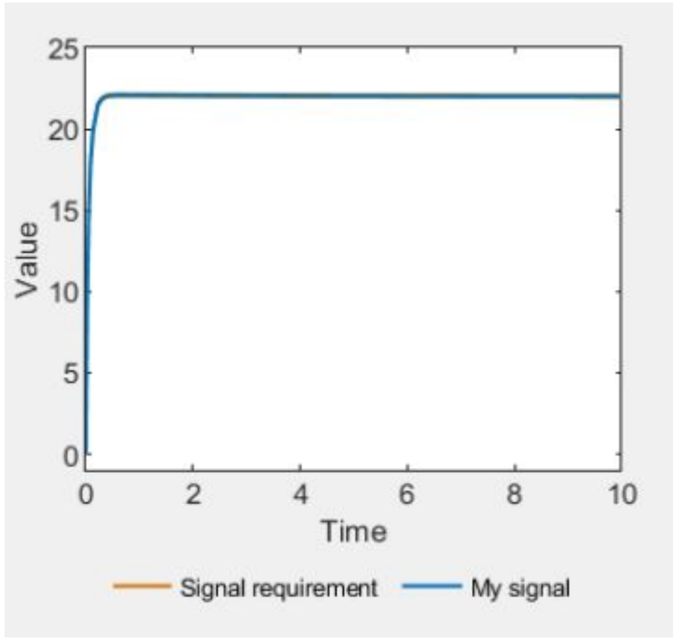
$y[k] = y_p[k] + y_i[k]$,

where:

$y_p[k] = K_p e[k]$

$y_i[k] = K_i T_s e[k] + y_i[k-1]$





(Time given in units of hours)

Ansys

ANSYS

Ansys is analysing software for mechanical product design, Fluid analysis and civil structure designs, it is widely used for developing analysing solutions in the industry.

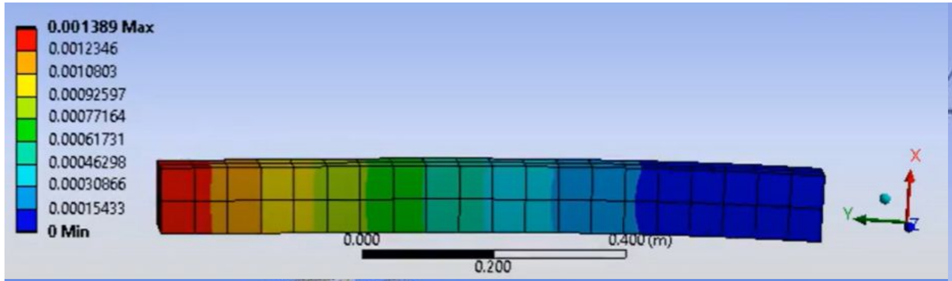
SOME NON TECHNICAL INFORMATION ABOUT ANSYS

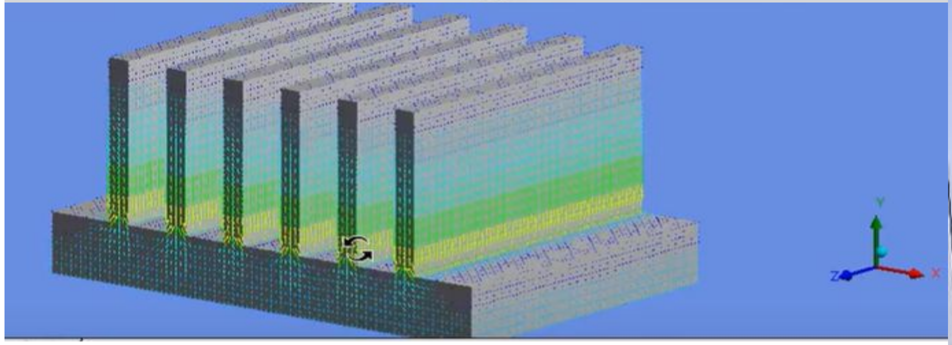
- Ansys, Inc. is an American company.
- It develops and markets CAE/Multiphysics engineering simulation software
- Ansys was founded in 1970 by John Swanson
- Ansys went public on NASDAQ in 1996 and is now component of the NASDAQ-100

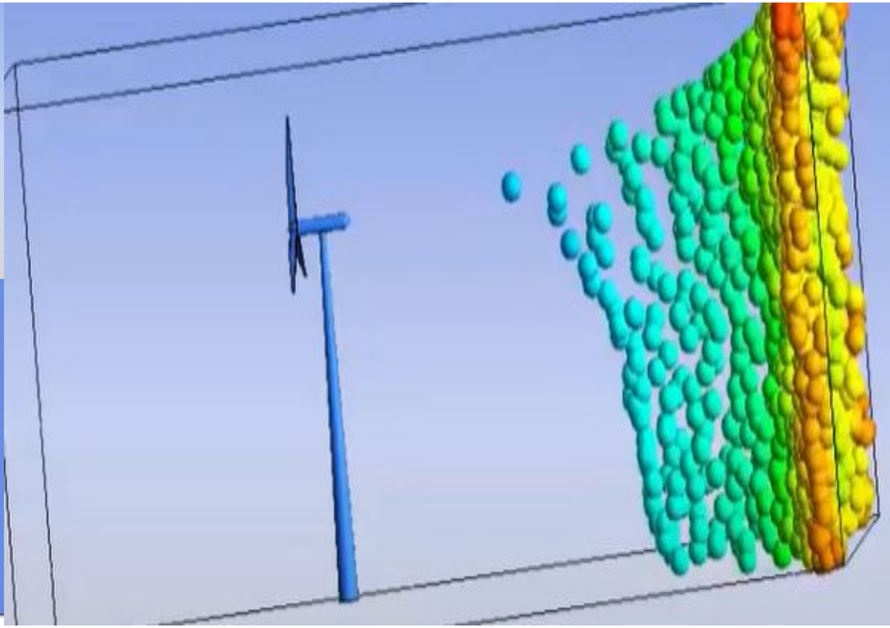
TYPES OF ANALYSIS

We can perform mainly 3 types of operations on Ansys

- Structural Analysis
- Thermal Analysis
- Computational Fluid Dynamics



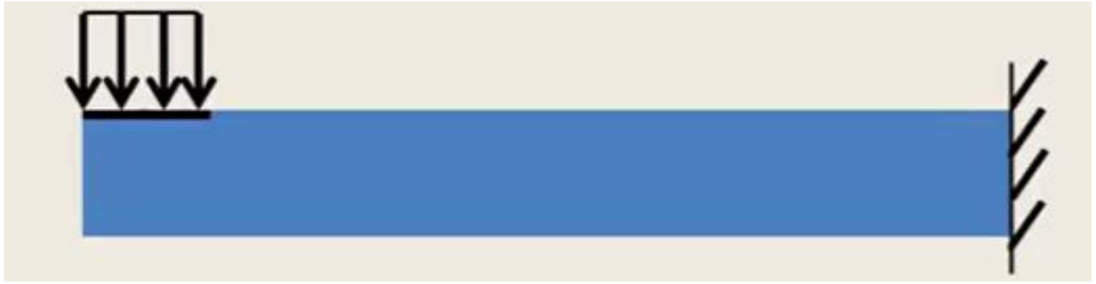




STRUCTURAL ANALYSIS

Structural analysis is the process of calculating and determining the effects of **loads** and **internal forces** on a structure, building, machine part or any object.

Static analysis of a structure, building, machine part or any object is intended to calculate the effects of constant loads (constant with respect to time) on the structure ignoring the effects of inertia and shock that are commonly found when the applied loads change rapidly



SOME BASIC TERMS

Stress is the force acting on the unit area of a material.

Strain is the amount of deformation experienced by the body in the direction of force applied, divided by the initial dimensions of the body.

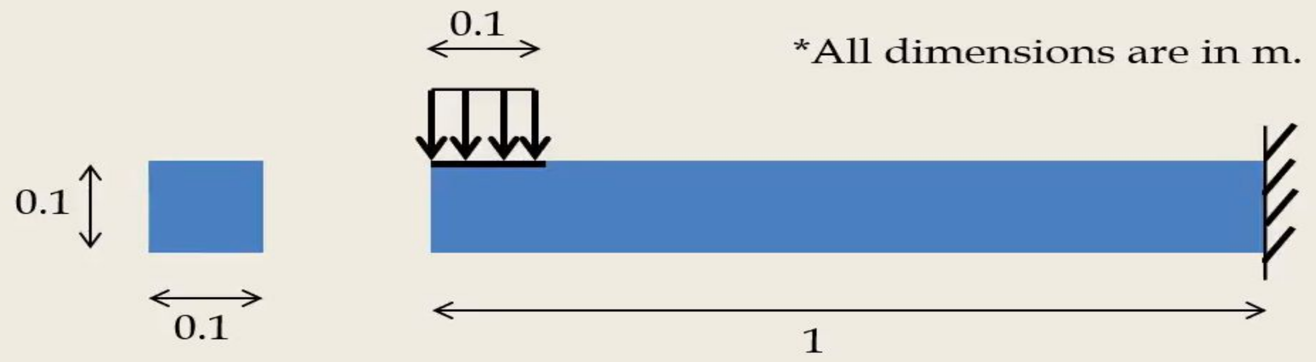
Young Modulus is a mechanical property that measures the tensile or compressive stiffness of a solid material when the force is applied lengthwise.

Bulk modulus (K or B) of a substance is a measure of how volumetrically resistant to compression that substance is. It is defined as the ratio of the infinitesimal pressure increase to the resulting relative decrease of the volume.

Shear modulus or modulus of rigidity, denoted by G, or sometimes S or μ, is a measure of the elastic shear stiffness of a material and is defined as the ratio of shear stress to the shear strain.

Poisson's ratio (nu) is a measure of the Poisson effect, the deformation (expansion or contraction) of a material in directions perpendicular to the specific direction of loading

ASSIGNMENT 1



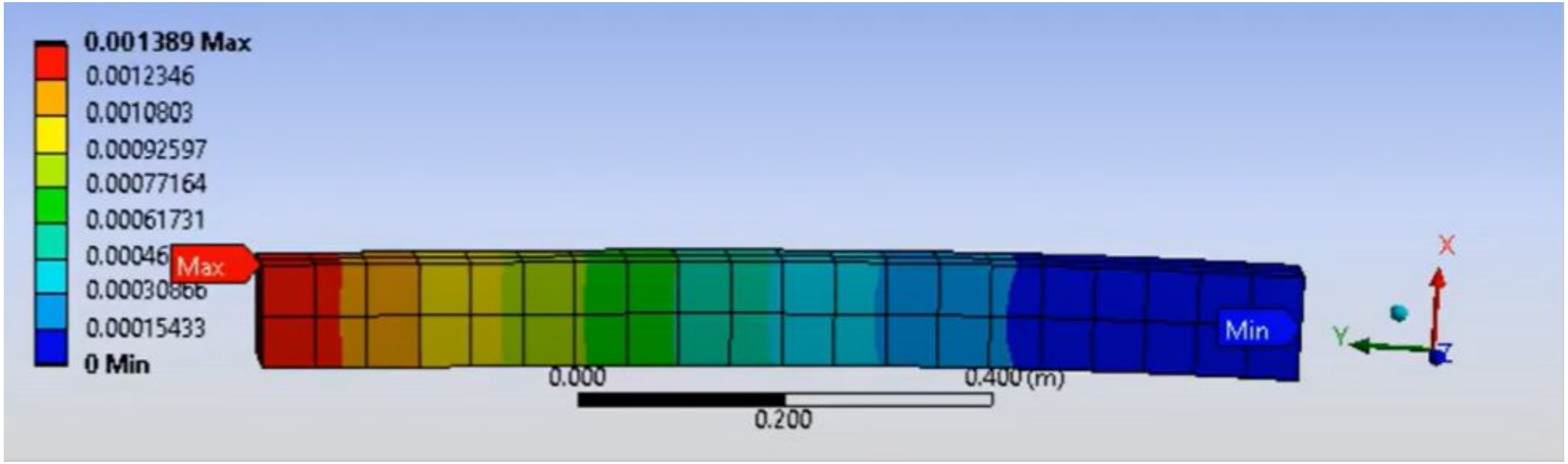
Youngs modulus of Wood = $\boxed{8\text{x}10^9 \text{ Pa}}$
Poisson's ratio of Wood = $\boxed{0.3}$

Weight of monkey ≈ 30 kg

Area on which monkey is sitting = 0.01 m^2

Pressure exerted on beam by monkey = 30x9.81/0.01
=29430 ≈ $\boxed{30{,}000 \text{ Pa}}$

RESULT



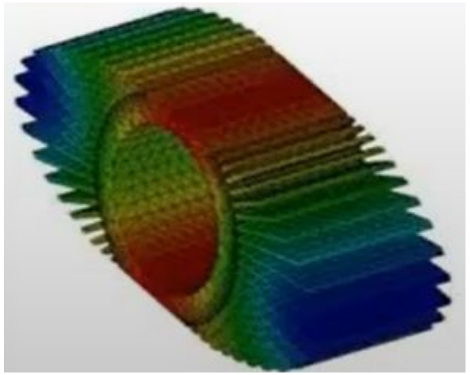
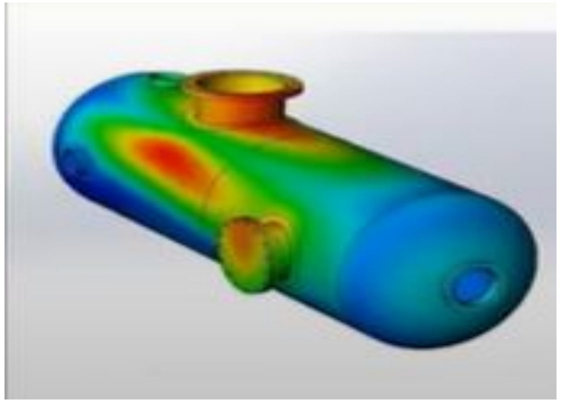
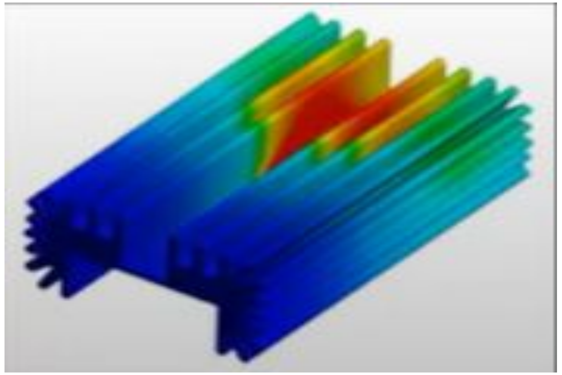
0.001389 Max
0.0012346
0.0010803
0.00092597
0.00077164
0.00061731
0.00046
0.00030866
0.00015433
0 Min
Max
Min
0.000
0.400 (m)
0.200
X
Y
Z

THERMAL ANALYSIS

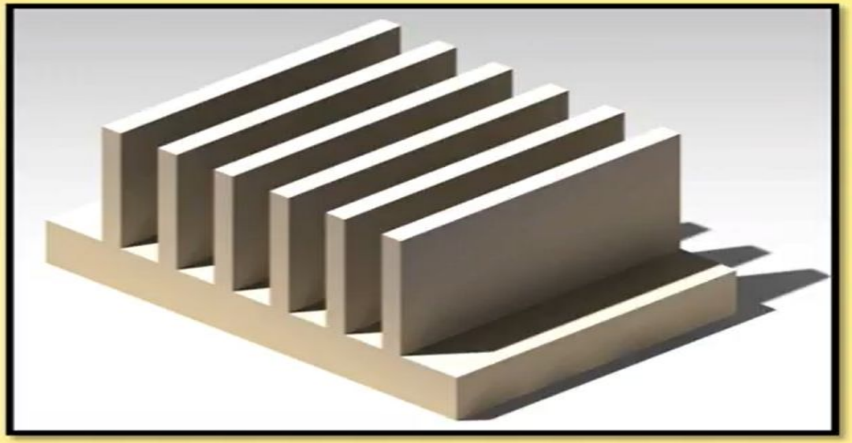
Thermal analysis is a general term defining a technique used to analyze the time and temperature at which physical changes occur when a substance is heated or cooled.

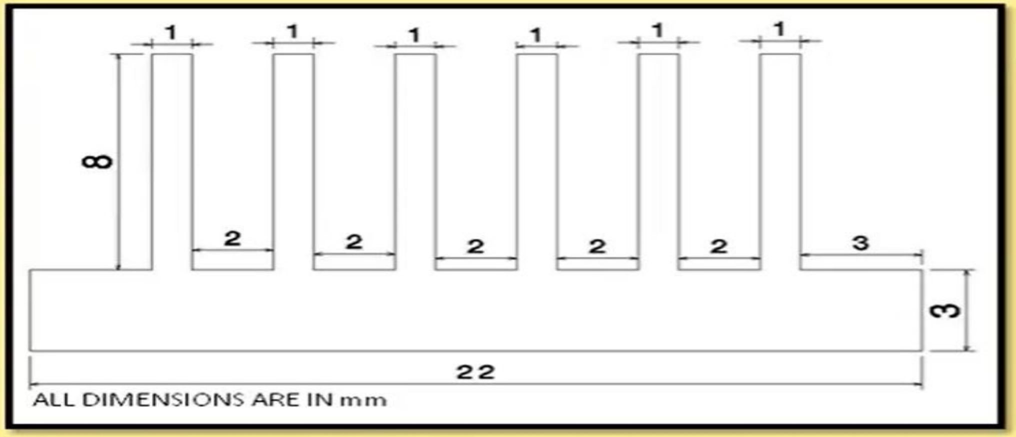
We perform thermal analysis to get

- Temperature Distribution
- Heat Flux distribution
- Structural Response Under Different Thermal Cooling Condition
- Thermal Stress

ASSIGNMENT 2





Input:

Material Aluminium

E = 70GPa

Poisson ratio = 0.3

Density =2800kg/m3

Thermal Conductivity =170 W/(m K)

Specific Heat = 870 J/(Kg K)

Thermal expansion Coefficient = 22*10e-6/oC

Boundary Condition:

Atmospheric Temperature = 28 ^{0}C

Heat Transfer Coefficient = 30 W(m$^{2\,0}$C)

Constant Heat Flux = 1000 W/m ^{2}

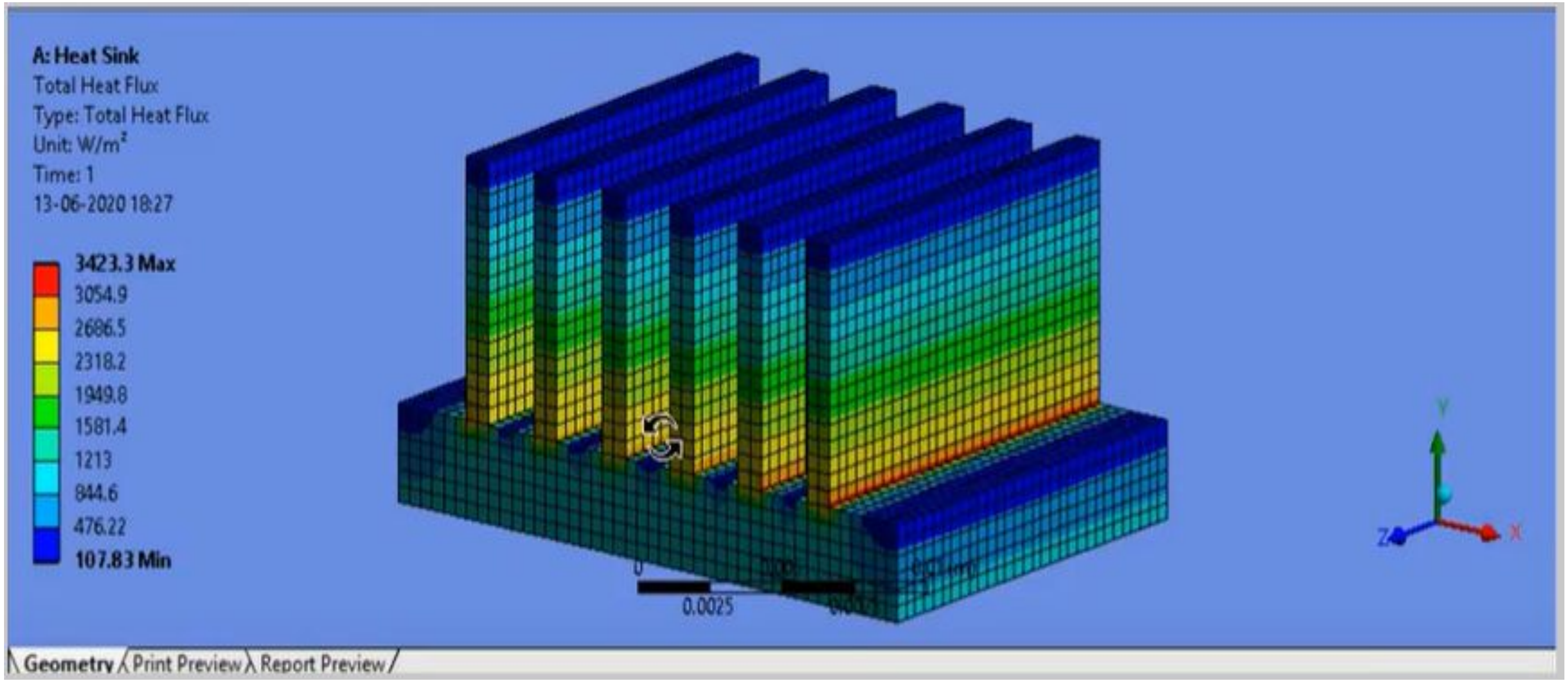
Result Using ANSYS :

Temperature = ???

Total Heat Flux = ???

Directional Heat Flux =????

RESULTS

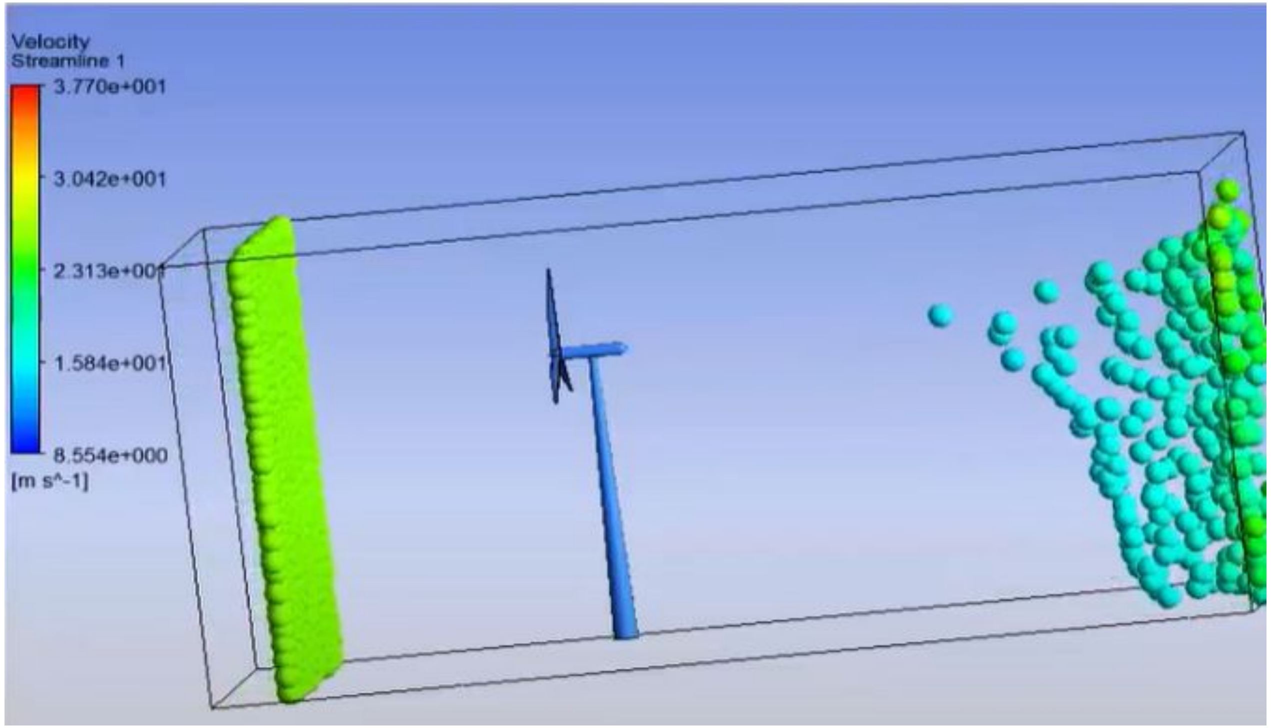

A: Heat Sink
Total Heat Flux
Type: Total Heat Flux
Unit: W/m²
Time: 1
13-06-2020 18:27
3423.3 Max
3054.9
2686.5
2318.2
1949.8
1581.4
1213
844.6
476.22
107.83 Min
0
0.0025
Y
Z
X
Geometry
Print Preview
Report Preview

COMPUTATIONAL FLUID DYNAMICS

Computational Fluid Dynamics(CFD) is a field in fluid dynamics that incorporates numerical analysis to simulate and solve problem involving fluid flows.

ASSIGNMENT 3

Create a cad file of wind mill and using Ansys show how it disturbs the smooth air flow
“Take help from the video tutorial provided”



SOME MORE STUFFS LEARNT

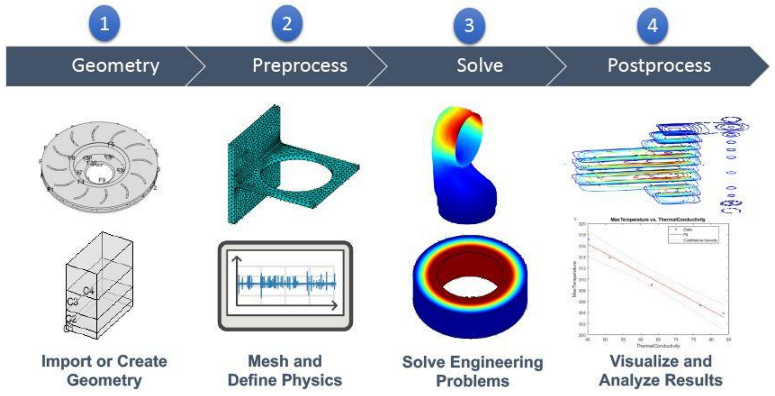
FEM (finite element method) (used by Ansys)

GItHub basics, to upload our solved assignment

FINITE ELEMENT METHOD

Finite Element Method(FEM) is a general numerical method for solving partial differential equations in two or three space variables

FEM can be used to analyze a wide range of solid mechanics problems including static, dynamic, buckling, modal analysis. It is also used in case of fluid mechanics, heat transfer and electromagnetics.


1
Geometry
2
Preprocess
3
Solve
4
Postprocess
Import or Create Geometry
Mesh and Define Physics
Solve Engineering Problems
Visualize and Analyze Results

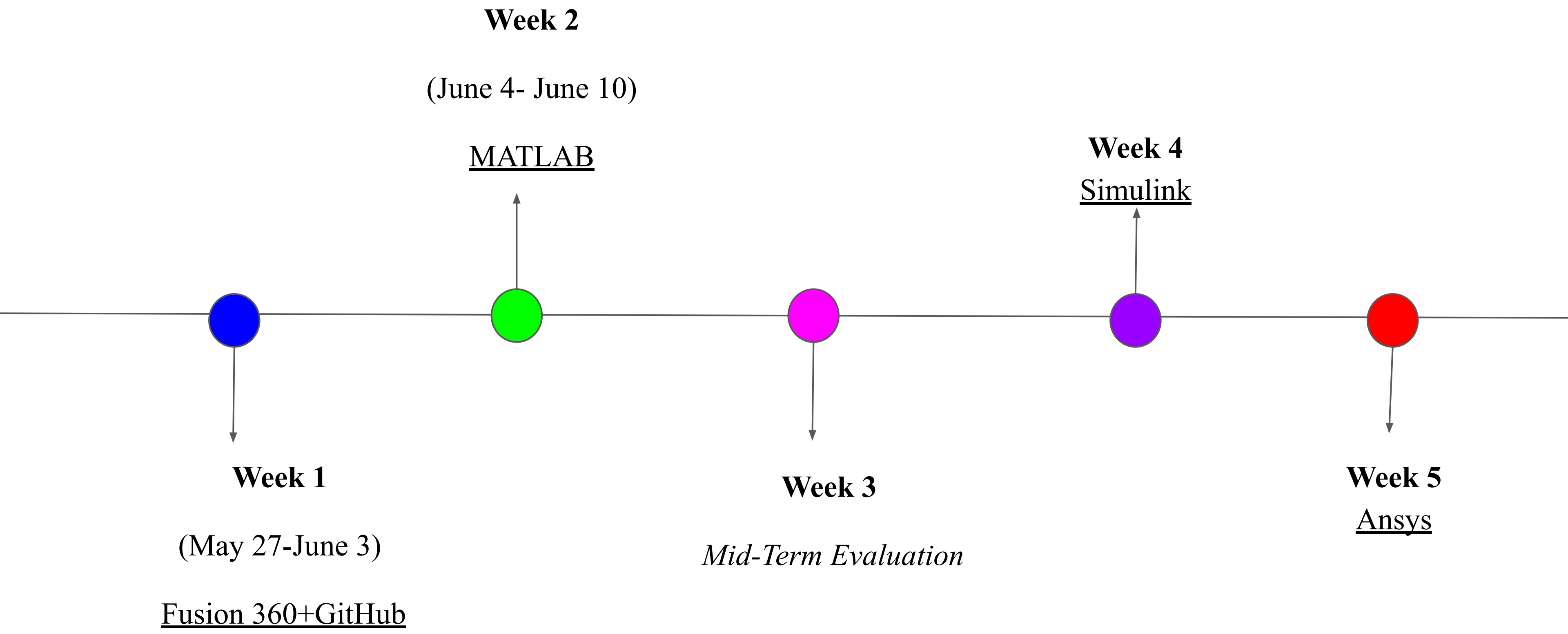
GITHUB

GitHub is a cloud-based service that helps developers store and manage their files, as well as track and control changes to their files.

GitHub offers a cloud-based Git repository hosting service.

In our project, we created a repository named Summer of Software and stored our tasks and assignments over there and it was easy for our mentors to track our progress.

Timeline of the Project



OUR TEAM

MENTORS

- Akarsh Raj
- Aryan Raj

MENTEES

Aakriti Gupta
Aarsha A Pillai
Alka
Anchit Gupta
Anil Yadav
Anuj Sarda
Arindom Bora
Arvind Mehta
Benison Kalyan Bage
Chitwan

Deepak K R
Devabrata Bothra
Dhara Sharma
Manas Singh
Manuja Pandey
Mayank Singh
Ritwik Shankar
Sanskar Sharma
Shreyash Nallawar
Sudhanshu Kumar
Suraj Singh Pate
Sonal S

Thank YOU