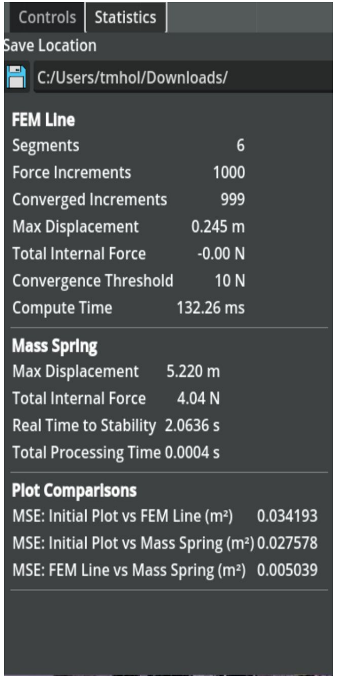
Team contributions

GitHub

Video

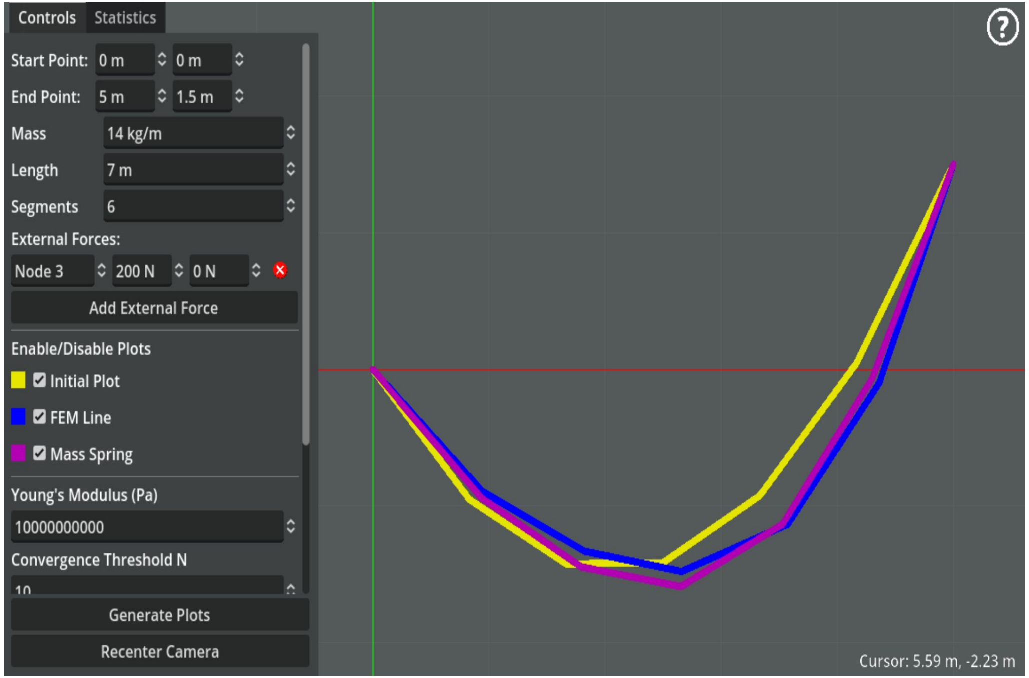


```
Save Location
C:/Users/tmhol/Downloads/

FEM Line
Segments                      6
Force Increments           1000
Converged Increments        999
Max Displacement        0.245 m
Total Internal Force    -0.00 N
Convergence Threshold      10 N
Compute Time          132.26 ms

Mass Spring
Max Displacement       5.220 m
Total Internal Force    4.04 N
Real Time to Stability  2.0636 s
Total Processing Time 0.0004 s

Plot Comparisons
MSE: Initial Plot vs FEM Line (m²)       0.034193
MSE: Initial Plot vs Mass Spring (m²) 0.027578
MSE: FEM Line vs Mass Spring (m²)     0.005039
```



Start Point: 0 m 0 m

End Point: 5 m 1.5 m

Mass 14 kg/m

Length 7 m

Segments 6

External Forces:

Node 3 200 N 0 N

Enable/Disable Plots

🟨 ☑ Initial Plot

🟦 ☑ FEM Line

🟪 ☑ Mass Spring

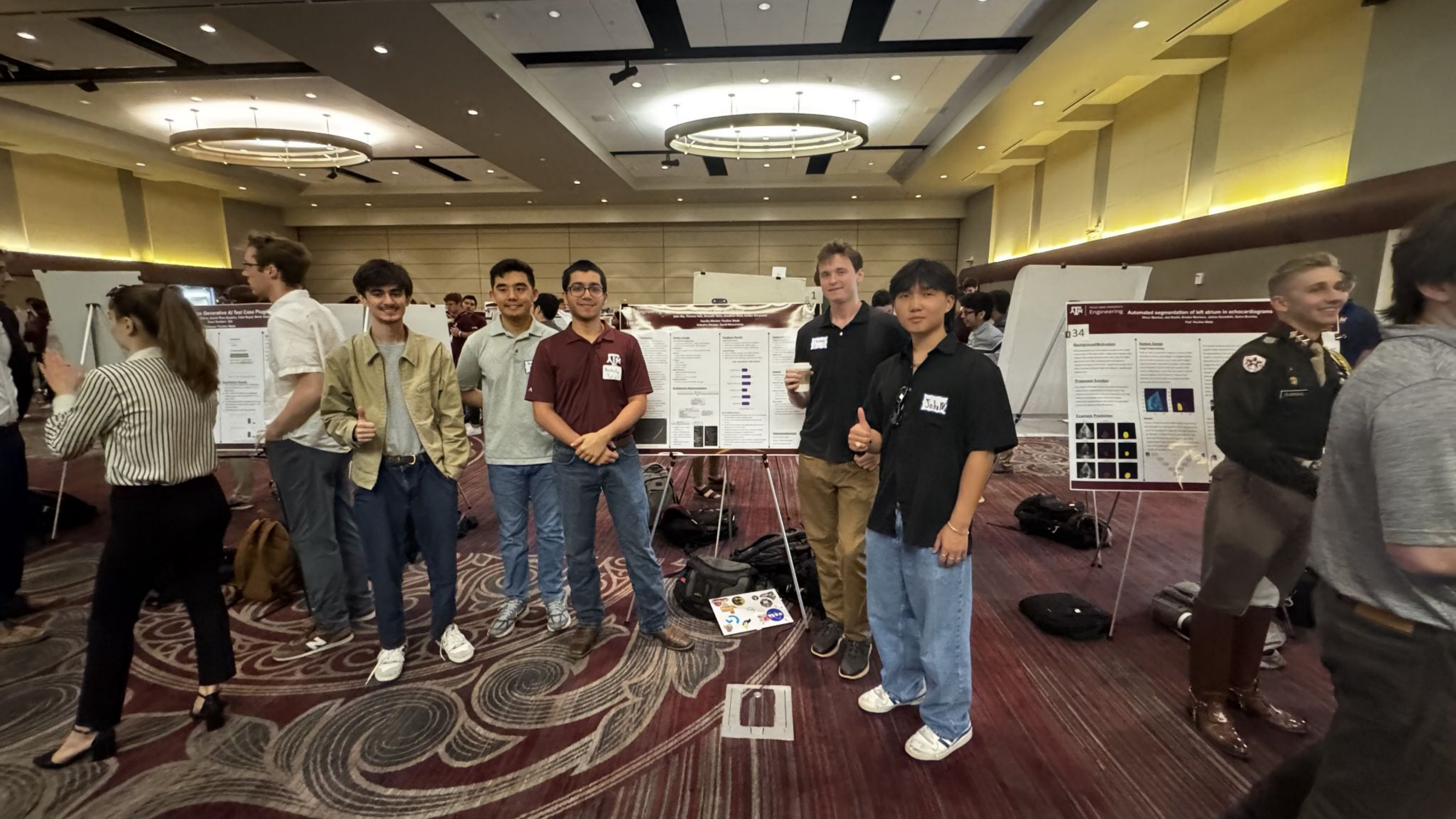
Young's Modulus (Pa)

10000000000

Convergence Threshold N

10

Cursor: 5.59 m, -2.23 m

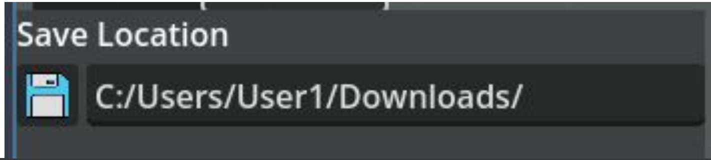

Generative AI Test Case Plugi
1
John W.
Engineering
Automated segmentation of left atrium in echocardiograms
34
Background/Motivation
System Design
Proposed Solution
Example Prediction

John Mo - peikaimo@gmail.com - linkedIn

Stakeholder Manager

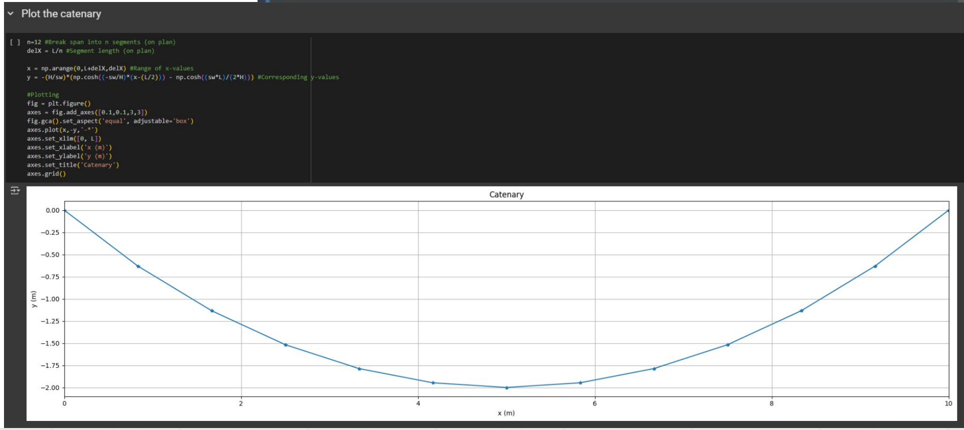
Research into python cable models

Worked with Jordan on save and exports



Save Location

C:/Users/User1/Downloads/



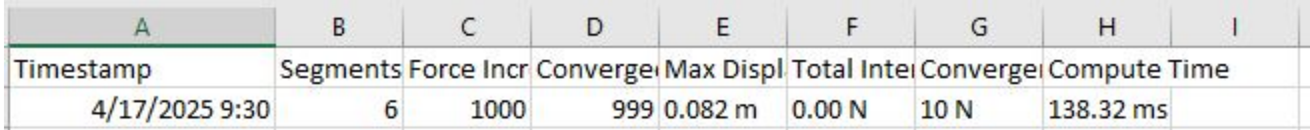
Plot the catenary

```
n=12 #Break span into n segments (on plan)
delX = L/n #Segment length (on plan)

x = np.arange(0,L+delX,delX) #Range of x-values
y = -(H/sw)*(np.cosh((-sw/H)*(x-(L/2))) - np.cosh((sw*L)/(2*H))) #Corresponding y-values

#Plotting
fig = plt.figure()
axes = fig.add_axes([0.1,0.1,3,3])
fig.gca().set_aspect('equal', adjustable='box')
axes.plot(x,-y,'-*')
axes.set_xlim([0, L])
axes.set_xlabel('x (m)')
axes.set_ylabel('y (m)')
axes.set_title('Catenary')
axes.grid()
```

Catenary
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−1.75
−2.00
y (m)
0
2
4
6
8
10
x (m)



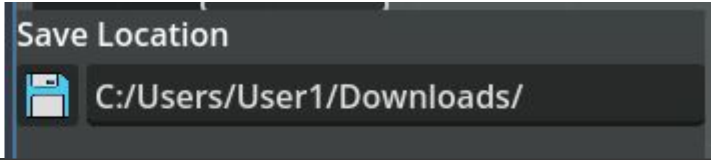
A	B	C	D	E	F	G	H	I
Timestamp	Segments	Force Incr	Converge	Max Displ	Total Inte	Converge	Compute Time	
4/17/2025 9:30	6	1000	999	0.082 m	0.00 N	10 N	138.32 ms	

Jordan - Jordandaryanani@gmail.com - Schedule Coordinator

Created Catenary Curve in Python

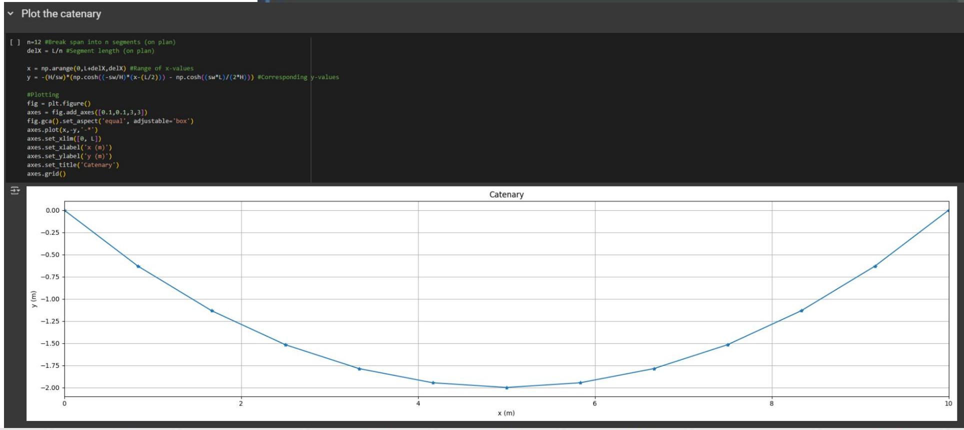
Created Save Location UI and Functionality

CSV File generation



Save Location

C:/Users/User1/Downloads/

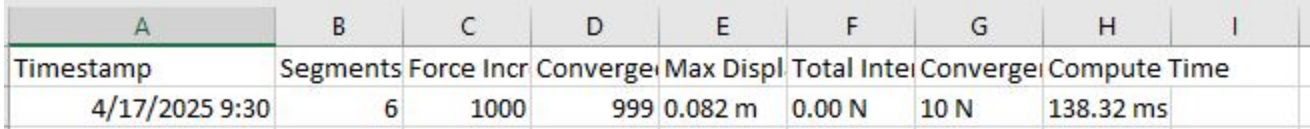


Plot the catenary

```
n=12 #Break span into n segments (on plan)
delX = L/n #Segment length (on plan)

x = np.arange(0,L+delX,delX) #Range of x-values
y = -(H/sw)*(np.cosh((-sw/H)*(x-(L/2))) - np.cosh((sw*L)/(2*H))) #Corresponding y-values

#Plotting
fig = plt.figure()
axes = fig.add_axes([0.1,0.1,3,3])
fig.gca().set_aspect('equal', adjustable='box')
axes.plot(x,-y,'-*')
axes.set_xlim([0, L])
axes.set_xlabel('x (m)')
axes.set_ylabel('y (m)')
axes.set_title('Catenary')
axes.grid()
```



A	B	C	D	E	F	G	H	I
Timestamp	Segments	Force Incr	Converge	Max Displ	Total Inte	Converge	Compute Time	
4/17/2025 9:30	6	1000	999	0.082 m	0.00 N	10 N	138.32 ms	

Thomas Holt - tmholt02@gmail.com - linkedin - (682) 319-4595

- Architecture and Coordination
- User Interface
  - Alerts
  - Background gridlines and controls
  - UI Components
- User input
  - User Control Node
- Integration of efforts
- Thread safety adaptations for progress readout
- Video

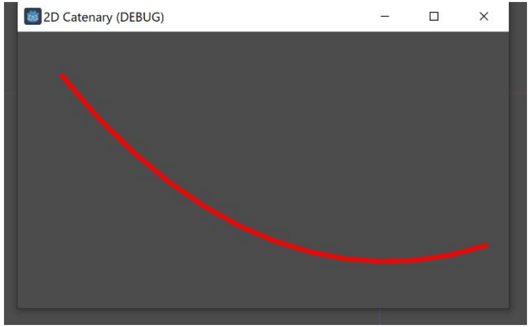
Mustafa - mustafa21tekin@gmail.com - Risk Manager

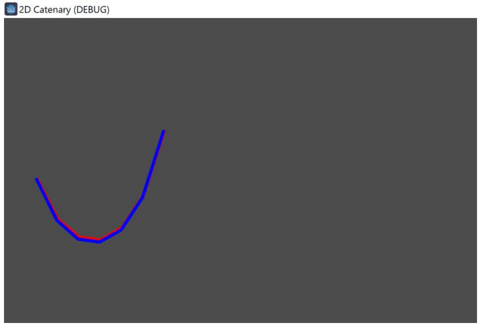
Asymmetric Initial Cable Curve (Godot/C#) - Left

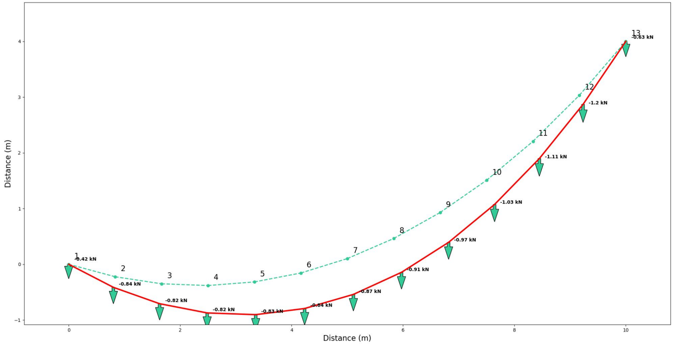
Post FEM Cable Curve (Godot/C#) - Middle

Initial Asymmetric Cable Curve & FEM Cable Curve (Python) - Right

Worked on Python Code & Converted it to C#





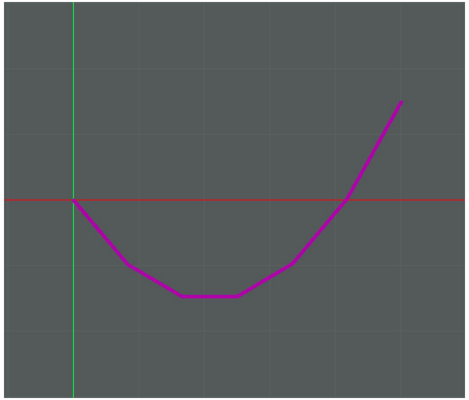


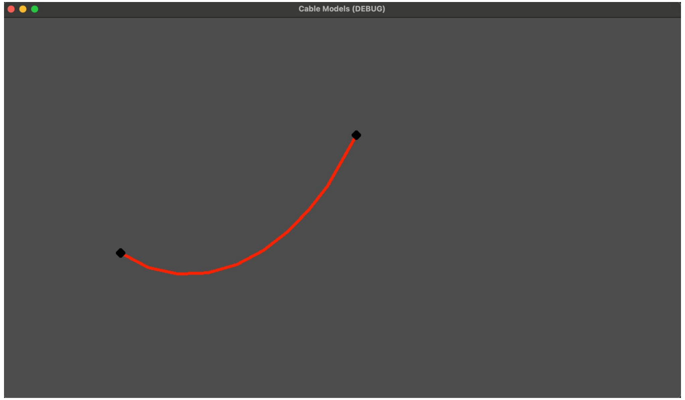
Jonathan - jonathanzhao64@gmail.com - Scope Manager

Studied cable models in python

Worked on the Mass Spring Implementation

Worked on a basic implementation of a real time cable





Cable Models (DEBUG)