

Create Sim2L - Pendulum Motion

<https://nanohub.org/tools/jupyter>

Dashboard

The screenshot shows the Jupyter Notebook tool page on nanohub.org. The page has a dark header with the nanohub logo and a 'MENU' button. Below the header, the title 'Jupyter Notebook' is displayed. A description states: 'Starts the Jupyter notebook server using the latest installed release of anaconda.' A blue button labeled 'Launch Tool' is highlighted with a red box. Below the button, it says 'Version 1.7 published on 27 Jan 2020', 'doi:10.21981/W6TE-1750 cite this', and 'This tool is closed source.' There is a link to 'View All Supporting Documents'. On the left, there are statistics: '2499 users, detailed usage', '0 Citation(s)', '2 questions (Ask a question)', '0 review(s) (Review this)', and '2 wish(es) (New Wish)'. At the bottom, there is a 'Share:' section with social media icons and a navigation bar with tabs: 'About', 'Usage', 'Citations', 'Questions', 'Reviews', 'Wishlist', 'Versions', and 'Supporting Docs'.

The screenshot shows the dashboard for Daniel Mejia on nanohub.org. The header includes the nanohub logo, the name 'Daniel Mejia', and a status 'Public Profile :: Your profile is currently public.' Below the header, there is a search bar with the text 'Search "jupyter notebook"'. A red arrow points from the 'Launch Tool' button in the previous screenshot to the search results. The search results are displayed in a table with columns 'Recent', 'Favorites', and 'All Tools'. The results include 'Jupyter Notebook' (highlighted in yellow), 'Jupyter Notebook (hub version 6.0)', 'Jupyter Notebook with Anaconda 2020.11', 'Jupyter Notebook with anaconda 5.1', 'Jupyter Notebook with anaconda4.1', 'Jupyter Notebook with anaconda4.4', 'Matlab Data Analysis Using Jupyter Notebooks', and 'Purdue ME 581 Numerical Methods in Engineering Using Jupyter Notebooks'. On the right side of the dashboard, there is a 'Recent' section with a list of groups: 'Gerhard Klimeck Research Forum', 'GPU Tool Developers', 'InstantOn Caching', 'NCN URE 2021', and 'NEMO5 distribution and support'. A '+ New Group' button is at the bottom right.

Create Sim2L - Pendulum Motion

The screenshot shows the nanoHUB Jupyter web interface. At the top, the nanoHUB and Jupyter logos are on the left, and 'Submit a ticket' and 'Terminate Session' buttons are on the right. Below the logos is a navigation bar with tabs: 'Files' (active), 'Running', 'Formgrader', 'Assignments', and 'Courses'. A message 'Select items to perform actions on them.' is displayed above a file list. The file list shows a folder named 'Gateways21Tutorial' with a 'Name' column header. Below the folder, a message states 'The notebook list is empty.' On the right side of the file list, there are buttons for 'Upload', 'New', and a refresh icon. The 'New' button is clicked, opening a dropdown menu. The menu is divided into two sections: 'Notebook:' and 'Other:'. Under 'Notebook:', there are options for 'Octave 4.2.0', 'Python (jul262020)', 'Python (tellurium)', 'Python 3', 'Python2', and 'R'. Under 'Other:', there are options for 'Text File', 'Folder', 'Terminal' (which is highlighted with a red rectangle), and 'noVNC Desktop'.

Submit a ticket Terminate Session

Files Running Formgrader Assignments Courses

Select items to perform actions on them.

Upload New

0 / Gateways21Tutorial Name

..

The notebook list is empty.



Notebook:

- Octave 4.2.0
- Python (jul262020)
- Python (tellurium)
- Python 3
- Python2
- R

Other:

- Text File
- Folder
- Terminal
- noVNC Desktop



Create Sim2L - Pendulum Motion

Submit a ticket

Terminate Session

```
clarksm@nanohub_1911864_32:~$ mkdir Gateways21Tutorial
clarksm@nanohub_1911864_32:~$ cd Gateways21Tutorial
clarksm@nanohub_1911864_32:~/Gateways21Tutorial$ git clone https://github.com/hubzero/gateways21tutorial.git
Cloning into 'gateways21tutorial'...
remote: Enumerating objects: 35, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 35 (delta 15), reused 22 (delta 5), pack-reused 0
Unpacking objects: 100% (35/35), done.
clarksm@nanohub_1911864_32:~/Gateways21Tutorial$
```

Submit a ticket

Terminate Session

Files

Running

Formgrader

Assignments

Courses

Select items to perform actions on them.



Upload

New ▾

↻

<input type="checkbox"/> 0 ▾	/ Gateways21Tutorial	Name ▾	Last Modified	File size
<input type="checkbox"/>	..		seconds ago	
<input type="checkbox"/>	gateways21tutorial		2 minutes ago	

Create Sim2L - Pendulum Motion

[Files](#) [Running](#) [Formgrader](#) [Assignments](#) [Courses](#)

Submit a ticket Terminate Session

Select items to perform actions on them.

Upload New ↕

☐ 0 ▾ / Gateways21Tutorial / gateways21tutorial / simtool


Name ▾

Last Modified

File size


..

seconds ago

☐  hubtoolsimtooltemplate.ipynb


4 minutes ago

2.34 kB

☐  pendulum.ipynb

4 minutes ago

6.57 kB

☐  pendulumDiagram.gif

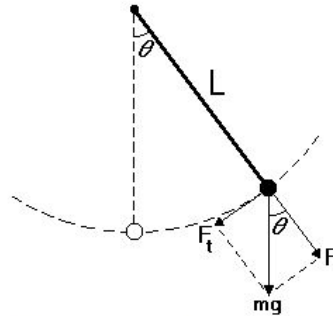
4 minutes ago

1.33 kB

Create Sim2L - Sim2L notebook

Problem Statement

Your assignment today is to write a Sim2L that solves a pair of ordinary differential equations describing the motion of a simple pendulum with damping applied.



The above diagram indicates the forces acting on a pendulum swinging from a fixed point. The equation describing the motion is:

$$\frac{d^2\theta}{dt^2} = -\frac{b}{m} \frac{d\theta}{dt} - \frac{g}{L} \sin\theta$$

where:

θ - angle measured from vertical (radian)

t - time (time)

b - damping coefficient (mass-radian/time)

m - pendulum mass (mass)

g - gravitational constant (mass/time²)

L - pendulum length (length)

The second order differential equation can be broken down into a pair of first order differential more suitable for numerical solution.

$$\frac{d\theta}{dt} = v$$



$$\frac{dv}{dt} = -\frac{g}{L} \sin\theta - \frac{b}{m} v$$

where:


v - angular velocity (radian/time)



It is suggested that the `solve_ivp` method from `scipy` be used to compute the numerical solution. Details of the `solve_ivp` method can be found online. The Sim2L results should provide enough information to plot position and velocity as functions of time.

Create Sim2L - Sim2L notebook

[Files](#) [Running](#) [Formgrader](#) [Assignments](#) [Courses](#)

Select items to perform actions on them. [Upload](#) [New](#) 

<input type="checkbox"/> 0 ▾	/ Gateways21Tutorial / gateways21tutorial	Name ▾	Last Modified	File size
	..		seconds ago	
<input type="checkbox"/>	bin		10 minutes ago	
<input type="checkbox"/>	data		10 minutes ago	
<input type="checkbox"/>	doc		10 minutes ago	
<input type="checkbox"/>	examples		10 minutes ago	
<input type="checkbox"/>	middleware		10 minutes ago	
<input type="checkbox"/>	simtool		4 minutes ago	
<input type="checkbox"/>	src		10 minutes ago	
<input type="checkbox"/>	 hubtoolsimtooltemplateExample.ipynb		10 minutes ago	2.24 kB
<input type="checkbox"/>	 pendulum.ipynb	Running	10 minutes ago	3.4 kB
<input type="checkbox"/>	README.md		10 minutes ago	68 B

Create Sim2L - workflow notebook

Standard Sim2L library imports

```
In [ ]: from simtool import findSimTools, searchForSimTool
        from simtool import getSimToolInputs, getSimToolOutputs, Run
```

Locate Sim2L

```
In [ ]: simToolName = "XXX"
```

For the selected Sim2L retrieve and list input parameters

This will provide the full list of parameters including descriptions, default values, and units.

```
In [ ]:
```

For the selected Sim2L retrieve and list output parameters

This will provide the full list of result parameters to be expected when running the Sim2L.

```
In [ ]:
```

Assign input parameter values

All input parameters should have default values. If you want to run non-default values set them here.

```
In [ ]:
```

Run Sim2L

Run the Sim2L and report the summary. The summary will indicate if any problems occurred during the run.

```
In [ ]: run = Run(simToolLocation, inputs)
        run.getResultSummary()
```

Extract results

Extract any results from the run needed to plot position and velocity versus time plots.

```
In [ ]:
```

Plot results

Using the plot library of your choosing plot position and velocity versus time for the pendulum. If you are not familiar with any plotting libraries *matplotlib* is simple and easy to use for static non-interactive plots.

```
In [ ]: import matplotlib.pyplot as plt
        %matplotlib inline
```