# JPCM (Japan Programming Cultivation Method)

Summer Python Bootcamp

PS1

Expected Due Date: May 17th, 2020 (Sun)

*Reminder:

There are three sections to this problem set: Beginner, intermediate, and advanced. Look through the tasks for each level and choose the one that you think you are capable of completing.

**Instructions for submission:

We will be using the Github Classroom platform to submit assignments conveniently plus to learn how to use Git commands. Since this is going to be your first problem submission using Github (hopefully) I will be giving you the steps of how to get started with your code and how to upload your completed assignment to the repository.

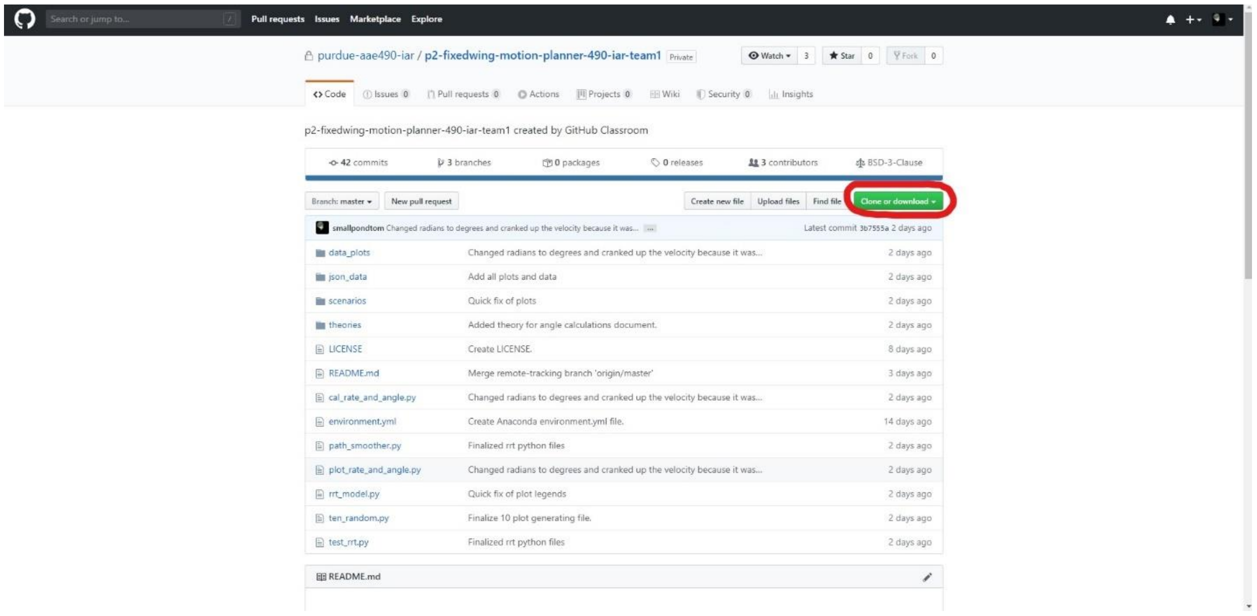
1. Download Pycharm

The first step is to download Pycharm on your computer. Follow the link for Pycharm https://www.jetbrains.com/pycharm/ . There are a bunch of YouTube videos that explain how to do this so check those out. The caveat is to add Pycharm to your path.

2. Navigate to a Folder to Clone the Repository

To start the problem set you will have to "clone" the repository and to do this you must first navigate yourself to a folder/directory that you want to save the files. These are the following steps on how to do that.

i. Open your Powershell/terminal by hitting the home button then searching for Powershell/terminal. For Windows users I recommend you use your Powershell instead of command prompt because Powershell understands bash scripting while the latter does not.
ii. On your Powershell/terminal navigate to the desired directory that you will save your problem set files into using the bash commands `cd` and `ls`. Read this link https://programminghistorian.org/en/lessons/intro-to-bash and master these basic bash commands. It is much better and powerful to be able to navigate around files using your Powershell/terminal
iii. Once you have navigated yourself to a desired directory, in your Powershell/terminal type `pycharm` and hit enter. This will open up Pycharm at the directory you have navigated to
iv. With Pycharm launched, on the bottom there is a tab called "terminal" that will open up the terminal console at the bottom of your IDE window
v. Go to the problem set repository page in Github Classroom.



purdue-aae490-iar / p2-fixedwing-motion-planner-490-iar-team1 Private

Watch 3 · Star 0 · Fork 0

Code · Issues 0 · Pull requests 0 · Actions · Projects 0 · Wiki · Security 0 · Insights

p2-fixedwing-motion-planner-490-iar-team1 created by GitHub Classroom

42 commits · 3 branches · 0 packages · 0 releases · 3 contributors · BSD-3-Clause

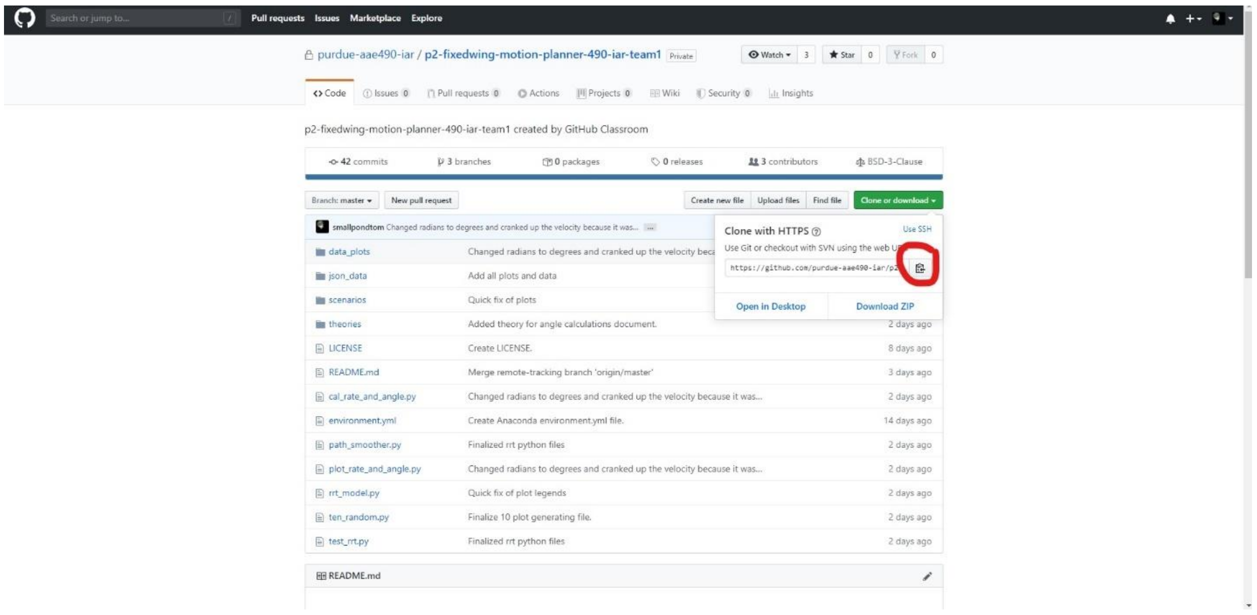
Branch: master · New pull request · Create new file · Upload files · Find file · Clone or download

smallpondtom Changed radians to degrees and cranked up the velocity because it was... · Latest commit 3b7555a 2 days ago

| | | |
|---|---|---|
| data_plots | Changed radians to degrees and cranked up the velocity because it was... | 2 days ago |
| json_data | Add all plots and data | 2 days ago |
| scenarios | Quick fix of plots | 2 days ago |
| theories | Added theory for angle calculations document. | 2 days ago |
| LICENSE | Create LICENSE. | 8 days ago |
| README.md | Merge remote-tracking branch 'origin/master' | 3 days ago |
| cal_rate_and_angle.py | Changed radians to degrees and cranked up the velocity because it was... | 2 days ago |
| environment.yml | Create Anaconda environment.yml file. | 14 days ago |
| path_smoother.py | Finalized rrt python files | 2 days ago |
| plot_rate_and_angle.py | Changed radians to degrees and cranked up the velocity because it was... | 2 days ago |
| rrt_model.py | Quick fix of plot legends | 2 days ago |
| ten_random.py | Finalize 10 plot generating file. | 2 days ago |
| test_rrt.py | Finalized rrt python files | 2 days ago |

README.md

Click the button circled in red.



purdue-aae490-iar / p2-fixedwing-motion-planner-490-iar-team1 Private

Clone with HTTPS · Use SSH

Use Git or checkout with SVN using the web U

```
https://github.com/purdue-aae490-iar/p
```

Then click the button circled in red to copy the url for the repository page.

vi. In your terminal, enter the following commands to clone the problem set repository to your current directory: git clone <url:link>. For the link, hit ctrl+V to paste the url that you have just copied. Now hit enter.

vii. Now you have successfully cloned the problem set repository to your desired directory.

3. Edit or make files and complete the problem set using Pycharm
4. Stage the files with Git

Now that you have finished your code and if it is working all fine, it is time for you to submit your work. The first step is to stage the work that you have done. In order to do this, in the terminal console of your Pycharm, enter the command `git add file1 file2 file3` ... (you can add as many files and directories as you want).

5. Commit your changes

Once you have staged your changes you have to commit them. To do this you will enter, `git commit -m “Edit files”.` In the quotation marks you can type any kind of message you want but the rule of thumb is to use present tense and not past tense to describe what you have done.

6. Push your local origin/master changes to remote master

Now that you have prepared every change you have made locally, you want to push all those changes to the remote master/repository which is the original repository you have cloned from in the beginning. For this enter the command `git push origin master`

Summary:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

C:\Users\Tomo> git --version
git version 2.23.0.windows.1
C:\Users\Tomo> ls


    Directory: C:\Users\Tomo


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----         5/1/2020  05:54 PM                .conda
d-----        4/25/2020  07:43 PM                .fontconfig
d-----        5/21/2019  05:23 PM                .ipython
d-----        3/23/2020  01:04 AM                .jupyter
d-----        10/6/2019  02:47 PM                .matplotlib
d-----        5/13/2019  07:17 PM                .PyCharmCE2019.1
d-----         6/6/2019  12:16 PM                .pylint.d
d-----         6/6/2019  10:31 AM                .vscode
d-r---        4/16/2020  08:17 AM                3D Objects
d-----         5/1/2020  05:48 PM                anaconda3
d-----       12/25/2019  02:59 AM                Apple
d-r---        4/16/2020  08:17 AM                Contacts
```

```
d-r---     5/3/2020 05:28 PM        Desktop
d-r---        4/16/2020  08:17 AM                Documents
d-r---         5/7/2020  10:22 PM                Downloads
d-r---        4/16/2020  08:17 AM                Favorites
d-----        5/10/2019  02:53 AM                Intel
d-r---        4/16/2020  08:17 AM                Links
d-r---        4/16/2020  08:17 AM                Music
d-----       12/21/2019  03:13 PM                New folder
dar--l         5/7/2020  09:42 PM                OneDrive
d-r---         5/5/2020  06:07 PM                Pictures
d-----         5/6/2020  09:28 PM                PycharmProjects
d-r---        4/16/2020  08:17 AM                Saved Games
d-r---        4/16/2020  08:17 AM                Searches
d-----         6/4/2019  08:42 PM                testCounter
d-----         6/4/2019  08:37 PM                testFolder
d-----        3/19/2020  07:44 PM                TomoCode
d-r---        4/25/2020  08:33 PM                Videos
-a----         6/6/2019  01:01 PM            137 .bash_history
-a----         5/1/2020  05:50 PM             43 .condarc
-a----       10/31/2019  01:34 PM             58 .gitconfig
-a----       11/15/2019  10:23 PM           7680 EXCEL.box
-a----        10/7/2019  02:41 AM          15428 RefEdit.exd


C:\Users\Tomo> cd .\PycharmProjects\
C:\Users\Tomo\PycharmProjects> ls


    Directory: C:\Users\Tomo\PycharmProjects


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----         5/7/2020  04:47 AM                p2-fixedwing-motion-planner-490-iar-
team1
d-----         5/1/2020  08:21 PM                PythonRobotics
d-----         9/3/2019  09:59 PM                random_project


C:\Users\Tomo\PycharmProjects> git clone https://github.com/serpapi/google-search-
results-python.git
Cloning into 'google-search-results-python'...
remote: Enumerating objects: 184, done.
remote: Counting objects: 100% (184/184), done.
remote: Compressing objects: 100% (101/101), done.

Receiving objects: 100% (456/456), 126.45 KiB | 3.24 MiB/s, done.
Resolving deltas: 100% (260/260), done.
C:\Users\Tomo\PycharmProjects> ls


    Directory: C:\Users\Tomo\PycharmProjects


Mode                LastWriteTime         Length Name
```

```
----                -------------         ------ ----
d-----         5/8/2020  07:33 PM                google-search-results-python
d-----         5/7/2020  04:47 AM                p2-fixedwing-motion-planner-490-iar-
team1
d-----         5/1/2020  08:21 PM                PythonRobotics
d-----         9/3/2019  09:59 PM                random_project


C:\Users\Tomo\PycharmProjects> cd .\google-search-results-python\
C:\Users\Tomo\PycharmProjects\google-search-results-python [master ≡]> pycharm

C:\Users\Tomo\PycharmProjects\google-search-results-python [master ≡]> git add
file1.py file2.py dir.py
C:\Users\Tomo\PycharmProjects\google-search-results-python [master ≡]> git commit -m
“Edit files”
C:\Users\Tomo\PycharmProjects\google-search-results-python [master ≡]> git push
origin master
```

***Deliverables:

For all levels, turn in your

1. source code
2. plots
3. A word document showing your output for all test cases

## Beginner

For this fundamental level, we will start off by understanding the basics of python programming. Namely, we will be concentrating on loops and functions.

TASK1:

Create a function that takes in an input of a positive integer and creates a pyramid of stars like the following.

If the input N=6, meaning a pyramid 6 stories tall.

```
*
**
***
****
*****
******
```

TASK2:

Now building on TASK1, we will create a function that takes in the number of stories for the Fibonacci Number Pyramid as an input and prints out the Fibonacci Number Pyramid for the corresponding story. For example, if the input is N = 4, the output would be

```
1

1 1

1 2 1

1 3 3 1
```

TASK3:

For this task you will have to create a function that takes in a n by n matrix list[list[]] that spits out the inverse of the matrix as an output. You can NOT use any modules for this; that is, you MUST NOT use numpy.

Test your code with the following matrices

$$\begin{bmatrix} 2 & 7 \\ 4 & 15 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 8 & 11 \\ 8 & 6 & 0 \\ 3 & 22 & 5 \end{bmatrix}$$

## Intermediate

For the intermediate level we will concentrate on plotting and data manipulation. You will master matplotlib.

TASK1:

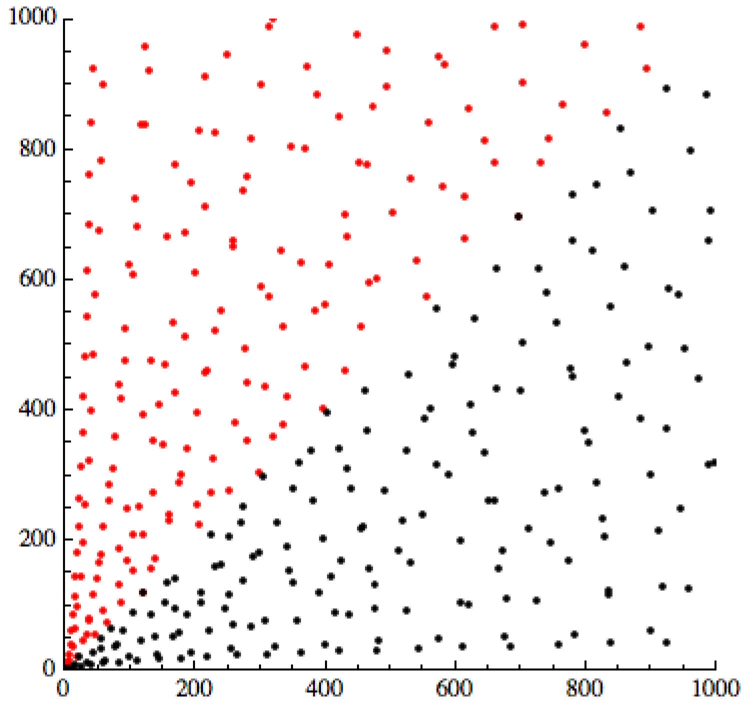
Create a function that takes in the input of a positive integer. The function will find all the combinations of the Pythagorean Triples up to the input number. For example, if the input is N=20, the output will be the following.

N = 20

Output = [[3, 4, 5], [6, 8, 10], [5, 12, 13], [8, 15, 17], [9, 12, 15]]

TASK2:

Create another function that plots the scattering of all the Pythagorean Triple legs in one plot corresponding to the input number (positive integer number) or the outputs for the function you have created in TASK1. More specifically if the Pythagorean equation is $a^2 + b^2 = c^2$, plot (a, b). The plot will look like the following.



TASK3:

Elaborating on TASK2, expand the plot to the entire plane, (+a, +b), (-a, b), (a, -b), and (-a, -b) and not only for positive integers.

TASK4:

Now extend TASK3 to a 3-dimensional plot by including the hypotenuse, (±a, ±b, ±c).

TASK5:

Now we want to integrate everything we have done so far. Create a function that takes in a target integer (positive or negative) and another integer that indicates the search limit. If the target integer T=5 and search limit integer N=20, the output is going to be [[3, 4, 5], [6, 8, 10], [9, 12, 15]]. Let me explain this. The first combination is the array that contains the target integer 5, and the other two arrays are the multiples of the first array, [6, 8, 10] = 2*[3, 4, 5] and [9, 12, 15] = 3*[3, 4, 5] and all numbers are below the search limit of 20. And plot the following on one plot (a, b), (a, c), and (b, c).

Test your code with the numbers:

N = 5, T = 20

N = 8, T = 100

N = 13, T = 1000

## Advanced

For this level, I will ask you for more than the other two levels, and for this problem set we will be concentrating on building classes.

TASK1:

Create a class inside one python file that numerically integrates first order differential equations using the Euler Method. Do NOT use any modules that does the Euler method for you, you must build one from scratch. You must determine what is going to be the input and what is going to be the output by yourself. For results return the list of y values for corresponding t-span and plot the (y vs t) plot.

Test it with the following equation.

$$\frac{dy}{dt} = tany + siny + 1$$

TASK2:

Create a class in a separate python file for 4th order Runge Kutta Method that numerically integrates first order differential equations. Same requirements as TASK1, and for results return the list of y values for corresponding t-span and plot the (y vs t) plot.

Test it with the following equation.

$$\frac{dy}{dt} = tany + siny + 1$$

TASK3:

Repeat the same thing but next using Gill's 4th order method.

Test it with the following equation.

$$\frac{dy}{dt} = tany + siny + 1$$

TASK4:

Now create a function that calculates the error between all 3 methods for the y-values with equal timespans. Next plot the errors over t-span.

$$error = y_{i,\mathrm{k}} - \frac{y_{\mathrm{i},1} + y_{\mathrm{i},2} + y_{\mathrm{i},3}}{3}$$

Where for subscript k

- k=1 $\Longrightarrow$ Euler method
- k=2 $\Longrightarrow$ 4th order Runge Kutta
- k=3 $\Longrightarrow$ Gill's 4th order method

Test it with the following equation.

$$\frac{dy}{dt} = tany + siny + 1$$

*When using the class objects create a separate execution file to do all the testing and call it test.py.