Lab Assignment 5: Mercury's Perihelion Precession and General Relativity

According to Wikipedia, general relativity "is the geometric theory of gravitation published by Albert Einstein in 1915 and is the current description of gravitation in modern physics." In this lab assignment, a student completes a Python program to test with data an accurate prediction of Einstein's theory, namely the perihelion precession of Mercury. Mercury's orbit around the Sun is not a stationary ellipse, as Newton's theory predicts when there are no other bodies. With Einstein's theory, the relative angle of Mercury's perihelion (position nearest the Sun) varies by about 575.31 arcseconds per century.

Version 0: Get Started

Unzip files in the `V0GetStarted.zip` file into your Working Directory. The `horizons_results.txt` file, which you may open and review in Spyder, is a relatively large *text* file of solar system data, called ephemerides, downloaded from a NASA Horizons Web Interface using the following parameters.

Parameter	Setting
Ephemeris Type:	**Vector Table**
Target Body:	**Mercury** [199]
Coordinate Center:	**Sun (body center)** [500@10]
Time Specification:	Start=**1800-01-01**, Stop=**2000-12-31**, Step=**1 d**
Table Settings:	quantities code=**1**; output units=**KM-S**; labels=**NO**; CSV format=**YES**; object page=**NO**
Download Results:	**horizons_results.txt** (plain text file)

Open the `perihelion.py` file in the Editor. The program consists of one `main` and seven additional functions. Run the program. As it runs, it outputs a few lines of text to the Console to indicate progress. It also outputs a file, `horizons_results.png`, in the Working Directory, and a figure under Plots. If the program crashes, verify that the `horizons_results.txt` file is in the Working Directory.

Open the `horizons_results.png` file outside Spyder (right-click and select "Open externally"). It is a *binary* file version of the figure in the Plots pane. Compare it to the `horizons_results_v0.png` file from the `V0GetStarted.zip` file. The two `.png` files should look (and even be exactly) identical.

Applying your knowledge and skill, analyze the `main` and the additional functions of the `perihelion` program. Use the debugger to help. Examine, with the Variable Explorer, the `list` structure, `data`, after each of the `loaddata`, `locate`, `select`, and `makeplot` functions is invoked by `main`.

Version 1: Load and Select

Unzip the one file, `horizon_results_v1.png`, in the `V1Load&Select.zip` file into your Working Directory. The output file (and associated figure), `horizons_results.png`, of your `perihelion` program, when you complete Version 1 sufficiently, should look like this file when opened.

Implementations of `loaddata`, `num2dict`, and `select` simulate Version 1 requirements to help Get Started. Complete them as follows without altering other code or the data file. After adding a second argument, rename `num2dict` to `str2dict`. Invoke it, in `loaddata`, as "`str2dict(num, line)`".

A correct `str2dict` function extracts a numeric date, a `float` (e.g., `2378496.5`), a string date, a `str` (e.g., `"1800-Jan-01"`), and 3D coordinates, a `tuple` of `floats`, from a line of text, a `str`, passed to it when invoked. Return values as a `dict` with keys `'numdate'`, `'strdate'`, and `'coord'`, respectively. Correct `str2dict` incrementally. When done, eliminate the `num` argument and associated code.

The `select` function returns an output argument, a `list`, with selected entries of its input one, also a `list`. Correct it to use inputs `ystep`, an `int`, and `month`, a `tuple` of `str`, to select the entries whose `'strdate'` specifies a year that is an integer multiple of `ystep` and a month that is in `month`.

Don't forget additional *implied* Version 1 functional requirements. Some annotations, i.e., *x*- and *y*-axis labels, are missing in the file and figure outputs of the Version 0 program. Edit your Version 1 program to include them *functionally*, thinking *non-functionally* where the best place is to include them.

Write suitable comment headers, in *your own words*, for the `loaddata`, `str2dict`, `locate`, and `select` functions. A function's comment header should summarize its purpose, arguments, and side effects, such as Console input and output, plots included, and file input and output. Paying attention especially to the reported percentages, complete the initial comment header of the program.

Submit your Version 1 solution via eClass by the Version 1 deadline. Submit the `perihelion.py` file only. Before submission, test it in a folder where all other relevant files are as originally given.

Version 2: Save and Refine

Unzip the `V2Save&Refine.zip` file into your Working Directory. Each `horizons_results_YYYY-MMM-DD.txt` file, which you may open and review in Spyder, is a relatively small file downloaded from NASA Horizons with the same parameters as before except for the Time Specification (see below). This data is given at half-century intervals, not at the quarter-century intervals Version 1 would imply.

Download Results (`+.txt`)	Time Specification
`horizons_results_`**`1800-Mar-21`**	`Start=`**`1800-03-20`**`, Stop=`**`1800-03-22`**`, Step=`**`1 m`**
`horizons_results_`**`1850-Jan-28`**	`Start=`**`1850-01-27`**`, Stop=`**`1850-01-29`**`, Step=`**`1 m`**
`horizons_results_`**`1900-Mar-04`**	`Start=`**`1900-03-03`**`, Stop=`**`1900-03-05`**`, Step=`**`1 m`**
`horizons_results_`**`1950-Jan-11`**	`Start=`**`1950-01-10`**`, Stop=`**`1950-01-12`**`, Step=`**`1 m`**
`horizons_results_`**`2000-Feb-16`**	`Start=`**`2000-02-15`**`, Stop=`**`2000-02-17`**`, Step=`**`1 m`**

With Version 1, not only is the perihelion precession *inaccurate* but also it is *imprecise*. The *best fit* slope looks to be about 2000 arcsec/cent and the precession computed from *actual* data looks quite noisy. To address the imprecision, we refine the perihelia at half-century intervals. When sufficiently correct, the Version 2 program creates a `horizons_results.png` file that equals `horizons_results_`**`v2`**`.png`. It will also create a `horizons_results.csv` file, a spreadsheet of the plotted data. This file equals `horizons_results_`**`v1.csv`** before refinement and `horizons_results_`**`v2.csv`** after.

Add the statement "`savedata(data,'horizons_results')`" to the `main` function, immediately after the `makeplot` statement. Add a `savedata` function to output a *comma-separated-value* file, a text file (supported by Microsoft Excel), that writes the *first* argument, `data`, in the *format* implied by `horizons_results_`**`v2.csv`**. Extension `.csv` aside, the filename equals the *second* argument.

Edit the `select` invocation, in the `main` function, to *select* perihelia at half-century intervals. To *refine* the data, add the statement "`data = refine(data,'horizons_results')`" immediately after the `select` invocation. Implement the `refine` function, as summarized below, without modifying any other code or any data file. After refinement, the plot illustrates improved precision and accuracy. As above, the actual data used by the plot is saved to a text file accessible inside or outside Spyder.

The `refine` function returns a more precise version of its first input argument, `data`. Given a `.txt` file, like `horizons_results_1800-Mar-21.txt`, exists with a name equal to the second argument plus a suffix from the `'strdate'` of each `dict` entry in `data`, load each such file and locate perihelia. Return a `list` of `dict` entries, similar to `data`, where each entry is the *first* perihelion of each such file.

Don't forget *implied* Version 2 requirements. A title, which includes a slope of the best fit line, is missing in file and figure outputs of the Version 1 program. Edit your Version 2 program to produce it correctly. Use `r[0]`, the slope in arcsecs/day, in `add2plot`. There are 365.25 days per year, on average.

Write suitable comment headers, in *your own words*, for the `refine`, `makeplot`, `precess`, `add2plot`, and `savedata` functions. Review the initial comment header, especially reported percentages. Submit your Version 2 solution via eClass by the Version 2 deadline. Submit the `perihelion.py` file only. Before submission, test it in a folder where all other relevant files are as originally given.

Revision History

Created in 2021 by Dileepan Joseph and reviewed by Edward Tiong and Jason Myatt, this lab is inspired by a Dean's Research Award project (2018) by Ragur Krishnan and Joseph, a pre-pandemic opportunity that followed Krishnan's ENCMP 100 Programming Contest entry – Solar System Simulator (2017). With feedback from Wing Hoy, Joseph revised the lab in 2022, especially after a major course change.