LLA-to-ECEF Coding Challenge

SciTec, Inc.

Contents

1 Problem Statement

In the field of remote sensing, a common task that comes up is coordinate conversion. Two common coordinate systems are geodetic coordinates, consisting of latitude, longitude, and altitude (LLA) and geocentric coordinates, also called Earth-centered, Earth-fixed (ECEF). Descriptions of these coordinate systems as well as the transformations between them are described in `LLAtoECEF.pdf`.

In the package accompanying this document, the file `SciTec_code_problem_data.csv` contains data representing a time series of an object using LLA coordinates. It is your task to write a program that returns the ECEF velocity vector of the object at any requested time.

1.1 Expected Subtasks

- Read the CSV file
- Convert LLA to ECEF
- Calculate ECEF velocity at the time points given in the input file
- Interpolate ECEF velocity for any requested time

1.2 Calculating ECEF Velocity

The expected baseline for calculating ECEF velocities is the discrete $\Delta Position/\Delta Time$ between consecutive points. The velocity at the first input point is defined as 0. Velocity at times that don't coincide with points in the input file should be calculated by linearly interpolating the velocities calculated for input points. If you are using a more complex method for calculating or interpolating velocity, please note how and why you are doing it in your readme (it is not expected that you differ from the baseline).

2 Input

The input CSV file located in the package accompanying this document should be read in by your program. The CSV file consists of rows of comma-separated values, where each row contains the following fields:

- Time since the Unix epoch [seconds]
- WGS84 Latitude [degrees]
- WGS84 Longitude [degrees]
- WGS84 altitude [kilometers]

3 Output

Your program should call the function or class you created, and evaluate the ECEF velocity vector at two specific times: 1532334000 and 1532335268 (both given as seconds since the Unix epoch). The result should be printed to stdout for both of the given times.

4 What to Deliver

You should send back your source code (including build files/scripts if required), along with a readme that includes instructions for building and running your code.

5 Allowed Resources

Your program may be written in Python 3 (up to and including Python 3.6). You are encouraged to use online resources when completing the challenge; however you may not directly copy non-trivial pieces of code or solutions.

5.1 Python 3

Code may use Python builtins, as well as NumPy, SciPy, and Pandas. If using NumPy, SciPy, or Pandas, please specify what versions of them are installed in your environ-ment.