Instructions to reproduce the LoadDef results in Boehm et al. (2024, GJI):

1. **Download or clone the LoadDef software from the Github repository**: https://github.com/hrmartens/LoadDef
2. **Check out the "BoehmEtal2024" branch**.
3. **Activate the Anaconda environment** (per LoadDef instructions; see "**doc**" folder).
4. **Enter the BoehmEtal2024 folder**.
5. **Compute the load Love numbers (LLNs)** by running the "**run_ln.py**" script.
    a. The default is to compute LLNs for PREM ("**option = 1**" in the "SPECIFY USER INPUTS" section of the "run_ln.py" script). [Needed for reproducing Figure 8.]
        i. The script should run "out of the box" without any required changes.
        ii. To run at the command line, you can type: ***python run_ln.py***.
        iii. To use the multiple processors (for a speedier computation), you can instead type: ***mpirun -np X python run_ln.py***, where "**X**" should be replaced with the number of processors that you wish to use (e.g., 2, 3, 10).
        iv. For model details, see the BoehmEtal2024/input/Planet_Models/ folder.
        v. Note: Ensure you are in the **BoehmEtal2024 folder** when you run the script.
    b. After the PREM computation finishes, edit the "run_ln.py" script to compute LLNs for the gravitating homogeneous sphere example. To do this, set "**option = 2**" in the "SPECIFY USER INPUTS" section near the top of the script. This computation will be performed analytically. [Needed for reproducing Figures 6 and 7.]
    c. Next, repeat this procedure for the homogeneous sphere *without* gravity (set "**option = 3**"). Again, all inputs are already prepared; you simply need to set [option = 3] in the "SPECIFY USER INPUTS" section of the script. [Needed for Figure 7.]
    d. Output will be directed to LoadDef/output/Love_Numbers/LLN/
6. **Compute the load Green's functions (LGFs)** by running the "**run_gf.py**" script.
    a. The default is to compute LGFs for PREM. The script should run "out of the box" without any required changes. [Needed for Figure 8.]
        i. For example, with three processors: ***mpirun -np 3 python run_gf.py***
        ii. Note: Ensure you are in the BoehmEtal2024 folder when you run the script.
    b. Repeat this computation for the gravitating homogeneous sphere by uncommenting the next set of "lln_file" and "file_out" options in the "SPECIFY USER INPUTS" section. Run as above (in Step 6a). [Needed for Figures 6 and 7.]
    c. Repeat this computation for the non-gravitating homogeneous sphere by uncommenting the next set of "lln_file" and "file_out" options in the "SPECIFY USER INPUTS" section. Run as above (in Step 6a). [Needed for Figure 7.]
    d. Output will be directed to LoadDef/output/Greens_Functions/
7. **Compute load-induced displacements analytically** for 10-degree disks located at each pole and containing a uniform 1 meter of freshwater (1000 kg/m^3) throughout. For this analytical computation, use the "**run_an_caps.py**" script in the BoehmEtal2024 folder.
    a. The default will be to use the PREM LLNs. The script should run "out of the box" without any required changes.
        i. For example, with two processors: ***mpirun -np 2 python run_an_caps.py***
    b. Repeat for the gravitating homogeneous sphere (using analytically computed LLNs) by uncommenting the second "**pmod**" option in the "SPECIFY USER INPUTS" section. Leave all other inputs untouched. Run again. [Needed for Figure 6.]

   c. Repeat for the non-gravitating homogeneous sphere by uncommenting the third "**pmod**" option in the "SPECIFY USER INPUTS" section. Leave all other inputs untouched. Run again.
   d. Output will be directed to LoadDef/BoehmEtal2024/output/
8. **Generate grids for polar disk loads**, both with and without a linear taper at the edges, in preparation for estimating displacements via a convolution procedure.
   a. First, navigate into the "grids" folder (within the BoehmEtal2024 folder).
      i. From the BoehmEtal2024 folder: ***cd grids***
   b. Run "gen_cap_no-taper.py" with the default parameters. For example: ***python gen_cap_no-taper.py***
   c. Run "gen_cap_with-taper.py" with the default parameters. [Needed for Figures 7 and 8.] For example: ***python gen_cap_with-taper.py***
   d. Output will be directed to LoadDef/output/Grid_Files/nc/Custom/
   e. Navigate back up one directory. From the "grids" folder: ***cd ..***
9. **Compute surface displacements for symmetric polar disk loads** using a convolution approach by running the "**run_cn_caps.py**" script.
   a. The default settings are to compute the load-induced displacements for each of the two loads created in Step 8, assuming PREM structure, and adopting a station-centered convolution approach ("classic approach"). The script should run "out of the box" without any required changes. [Needed for Figure 8.]
   b. Repeat the step above for the *gravitating*, homogeneous sphere. To do this, you simply need to uncomment the "**pmod**" line corresponding to the gravitating homogeneous sphere model within the "SPECIFY USER INPUTS" section of "run_cn_caps.py". Run the code again. [Needed for Figure 7.]
   c. Repeat the step above for the *non-gravitating*, homogeneous sphere. To do this, you simply need to uncomment the "**pmod**" line corresponding to the non-gravitating homogeneous sphere model within the "SPECIFY USER INPUTS" section of "run_cn_caps.py". [Needed for Figure 7.]
   d. Output will be directed to LoadDef/output/Convolution/
   e. **Important note**: The computation can take several hours/days to complete, especially on a standard laptop computer, due to heavy refinement of the convolution mesh and the large number of observation points.
      i. The process can be sped up by coarsening the mesh, depending on the precision needs of the user. The mesh parameters are defined in the "SPECIFY USER INPUTS" section of "run_cn_caps.py".
      ii. The process can also be sped up by reducing the number of observation points, but the resulting solution curve may then not appear smooth. The observation points are defined in the "Lat_Profile.txt" file within the input/Station_Locations/ folder, located within the BoehmEtal2024 folder.
      iii. Furthermore, the process can be sped up by using a relatively new feature of LoadDef that takes a different approach to the convolution. Rather than define a separate convolution mesh for each station ("classic approach"), the new approach defines a single mesh that is common to all stations ("common-mesh approach").
         1. Examples of common meshes can be generated in the "grids" folder (within the BoehmEtal2024 folder): "caps_mesh_01.py" and "caps_mesh_02.py". Each script will generate the same common

mesh (using default parameters) using a slightly different procedure, and either can then be used in the "run_cn_caps.py" script.

2. To use the common-mesh approach, simply set the "common_mesh" flag to "True" within the "SPECIFY USER INPUTS" section of "run_cn_caps.py", and then uncomment the relevant pair of lines at the bottom of that code block (i.e., set the "convmesh" and "mesh_params" accordingly, based on the mesh parameters/name create in Step [9,e,iii,1] above).

iv. **Note**: A common-mesh approach is much faster but can be slightly less accurate due to a misalignment between the grid for the LGFs (symmetric about each observation point) and the grid used for the common mesh (symmetric with respect to each pole of the Earth). This subtlety will be explored further in due course; for now, please be aware of this.

f. **Recommendation**: Run "run_cn_caps.py" on several processors to complete the computation sooner. Note that each observation point (defined in "Lat_Profile.txt") is independent of the others and can therefore be run on a separate processor.

   i. Example, with ten processors: ***mpirun -np 10 python run_cn_caps.py***

10. **Combine the convolution output** (found under LoadDef/output/Convolution/; one file per observation point) into a single file for each Earth-model/load-model combination using "**run_combine_stations.py**", which is in the LoadDef/BoehmEtal2024/utility/ folder.
    a. Navigate into the "**utility**" folder (located within the **BoehmEtal2024 folder**).
    b. Run the script; for example: ***python run_combine_stations.py***
    c. Update the "**pmod**" parameter in the "SPECIFY USER INPUTS" section near the top of the script to select a different Earth model (and optionally update the "suffix" if using a different mesh or convolution approach), and then run again.
    d. Run the script for each of the three Earth models: (i) PREM, (ii) gravitating homogeneous sphere, and (iii) non-gravitating homogeneous sphere.
    e. Output will be directed to LoadDef/BoehmEtal2024/utility/output/

11. **Plot the results** using scripts in the "**plots**" folder (within the **BoehmEtal2024 folder**).
    a. Navigate into the "**plots**" folder (LoadDef/BoehmEtal2024/plots/).
    b. **To plot LLNs**: *python plot_ln.py*
       i. Note that you can select a different Earth model using the "pmod" parameter in the "SPECIFY USER INPUTS" section near the top of the script.
    c. **To plot LGFs**: *python plot_gf.py*
       i. Note that you can select a different Earth model using the "pmod" parameter in the "SPECIFY USER INPUTS" section near the top of the script.
    d. **To plot convolution results**: *python plot_cn_lat.py*
       i. Note that you can select a different Earth model using the "pmod" parameter in the "SPECIFY USER INPUTS" section near the top of the script.
       ii. You can also set the "taper" flag to "True" or "False", depending on whether you want to plot results with the linear taper applied to the load (taper=True) or with a sharp edge at the boundary of the disk load (taper=False).
    e. Output will be directed to LoadDef/BoehmEtal2024/plots/output/