**SimpleCode, LAMP, and LSTM via Python**

Contact for questions:

* Dayne Howard
  + dayneh@mit.edu
  + dhowar199@gmail.com
  + dancingdayne@gmail.com
  + 801-809-7758

**Background**

This document is meant to provide guidance on how to use several python scripts that were developed for training LSTM neural networks to map SimpleCode output to LAMP output. (See thesis: “Quantifying Extreme Event Statistics for Ship Motions and Loads Using Low-Fidelity Models and Recurrent Neural Networks”). A lot of time can be lost in creating data from SimpleCode and LAMP, making sure they line up with each other, and loading them into the LSTM training path correctly. Therefore, several scripts were created to streamline the process, and the documentation for these is contained here.

**Overview**

The diagram below shows the order that the scripts should be used and how their outputs are handled. The following sections describe each script in more detail and provide examples. The goal through this process is to provide the LSTM trainer with a set of SimpleCode motion/wave files and corresponding LAMP motion/wave files from which training, validation, and testing may be performed.

Diagram

Description automatically generated

**casesPy**

This is the script where the user will need to actually do some coding. The goal is to create a text file which will specify a set of simulation records to be used for training, validation, or testing. Below is an example of how the output should look (green for denoting what is actually in the file):

My description: Training records for my first experiment.

\*\*\*

SC

LAMP

\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5-0000001

\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5-0000002

\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa3.0\_s5-0000003

There are four important sections in the following order:

1. **The description**: This can be as many lines as you want, and serves as a reminder to you of what this data is for. Should denote what type of data it is (training/validation/testing) (do not mix these data together in the same file, make a separate casesPy file for each type of data for any given experiment).
2. **\*\*\***: Three stars denote the end of the description and start of handles.
3. **SCprefix** and **LAMPprefix**. These two lines will specify what will be attached to the front of the handles that follow. The first is for SimpleCode and the second is for LAMP. They must be unique. However, this enables the user to create experiments for other things like if they wanted to use a different version of LAMP they could specify “**LAMP2**”; or you could specify the hull that is being used in the experiment. “**LAMP.ONRFL**”. Don’t include any “**\_**” in the prefix.
4. **\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5-0000001**: The handles. “\_” mark the separation between each variable. The numbers can have decimals or not.
   * + h = primary significant wave height in METERS
     + p = primary wave modal period in SECONDS
     + a = primary sea heading (think “angle”) in DEGREES
     + hh = secondary significant wave height
       - * If the user wants to do a primary system only, then put 0 as the number after “hh”.
     + pp = secondary significant wave height
     + aa = secondary significant wave height
     + s = ship speed in KNOTS
     + -####### = the number for the realization or record. Must be SEVEN digits, corresponding to how SimpleCode produces output files.

The user can create experiments however they want, and at the end of the day they will only have to do this once. All of the other scripts will read from the created “casesPy.txt” file (or whatever you name it) for creating SimpleCode and LAMP input files, running them, and loading them into the LSTM trainer.

Within the script you will specify the experiments, the description, and the name of the text file to be created. Upon executing the script from the command line, the text file name and description will be confirmed with the user before creating the file (so that old file names don’t get inadvertently rewritten over.)

**SimpleCodePy**

When you run this script from the command line, you will be asked for a casesPy.txt file name. Once you enter it, this script will create SimpleCode input files for each handle in the list of casesPy.txt, and then run SimpleCode for those cases.

Outputs for SimpleCode will go wherever the SimpleCode.exe is located. There is a line of this script that specifies the command that will be used to run SimpleCode. The variable is called “command”. In its initial creation, it assumes SimpleCode.exe is in the same folder as SimpleCodePy.py. If you want to change this, then you’ll need to modify the script.

SimpleCode input files contain many other options and variables that are not specified by the handles, which only have primary/secondary wave characteristics, ship speed, and realization index. You can change those default values in the script in the “create\_SC\_input\_file” function.

**LAMPPy**

This is identical in function and features to SimpleCodePy except that it is for LAMP. Of note, LAMP input files can actually specify where the LAMP outputs are going. In the “create\_LAMP\_input\_file” function, there is a line that says, “!05 FOUT - Destination file for primary output”. Just beneath that, you can specify the output destination folder. I have it defaulted as “lamp\_output\_files”.

**File Structure**

To make life easy, I am going to specify a file structure with where each file should go and what they should be called. Names in < > are examples and will have varying names. Other file and directory names should be as spelled out here, otherwise alterations to the code will be needed. If it ends with a “\”, then that is a directory. Comments are text in Red.

* <my\_lstm\_folder\_name>\ This name doesn’t matter.
  + Cases\_files\
    - <my\_experiment\_cases\_training.txt>
    - <my\_experiment\_cases\_validation.txt>
    - <my\_experiment\_cases\_test.txt>

These are all created by casesPy.py

* + LAMP\_files\
    - <LAMP\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5-0000001.in>
    - <LAMP\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5-0000001.out>
    - <LAMP\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5-0000001.mot>
    - <LAMP\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5-0000001.vbm>
    - <LAMP\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5-0000001.wav>
    - <LAMP\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5-0000001.sea>

The .in, .out, .mot, .vbm, .wav, and .sea files are all created by LAMPPy.py

* + SimpleCode\_files\
    - <SC\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5.in>
    - <SC\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5-0000001.mot>
    - <SC\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5-0000001.vbm>
    - <SC\_h1.0\_p1.0\_a1.0\_hh1.0\_pp1.0\_aa1.0\_s5-0000001.wav>

The .in, .mot, .vbm, and .wav files are all created by SimpleCodePy.py

* + get\_motion.scpt this is a text file that tells lamp to generate .mot, .wav, .sea, and .vbm files
  + LMP\_ONRFL\_loads\_rigid.inp Used in LAMPPy.py
  + topsides\_fl.lmp Used in LAMPPy.py
  + flh.off Used in SimpleCodePy.py
  + fl154\_loads.inp Used in SimpleCodePy.py
  + <casesPy.py> The name of this can vary. You can have multiple of these.
  + LAMPPy.py Script name doesn’t matter, but there’s no reason to change it
  + SimpleCodePy.py Script name doesn’t matter, but there’s no reason to change it
  + SimpleCode.exe \*
  + lamp.exe \*
  + lmplot.exe \*
  + glut32.dll

\* LAMP and SimpleCode binaries should NOT be distributed outside the US Navy without permission. Questions related to the distribution of either code should be directed to the NSWC/CD Simulations and Analysis Branch (Code 851). Additionally, distribution must be through secure means (like DODsafe).