Convert Metocean data into a SACS seastate input file

C Kunte

November 8, 2022

Contents

1 Scope

The stand-alone script (Appendix A) converts Metocean data (Appendix B), into a readily usable SACS seastate input file (Appendix C).

The script is written in python scripting language, and it requires some user inputs. These can be keyed-in by editing the block of data within `BEGIN USER INPUTS` and `END USER INPUTS` in the script file `slc.py`, set pre-determined values (or defaults) as appropriate, and provide a comma separated data file (csv) from Metocean a the command line. The SACS seastate input file may be generated in two steps.

Generate a formatted csv file using the command. If the given Metocean data csv file name is, say, `test.csv`, then the formatted csv file name becomes `Ftest.csv`.

```
python3 fdf.py -f <csv file>
```

Then, generate a seastate file using the following command:

```
python3 slc.py -f <formatted csv file> > seastate.inp
```

```
$ python3 fdf.py --help
Format CSV file with Pandas
fdf.py 2022 ckunte

Usage: fdf.py (-f <file>)
       fdf.py --help
       fdf.py --version

Options:
  -h, --help   Show this help
  -f --file    Specify CSV input file to format (required)
```

```
$ python3 slc.py --help
Generate SACS storm load cards from a CSV file
slc.py 2022 ckunte
Tested for python v3.8.10, v3.10.8 w/ pandas >= v1.5.1

Usage: slc.py (-f <file>)
       slc.py --help
       slc.py --version

Options:
  -h, --help   Show this help
  -f --file    Specify CSV input file (required)
```

1.1 Limitations and workarounds

The limitations of the script are (a) only the system memory allocated for the script, and (b) SACS's four character field for load cases. In other words for load cases greater than 9999, the load case numbering will default back to 0000. This can be overcome by splitting the data file to be less than 9999 load cases.

1.2 Requirements

The script requires the following and can be run at command line interface with the following installed and available:

1. `python` v3 scripting language,
2. `pandas` data analysis library,[1]
3. `docopt` command-line interface description language.[2] .

[1] Install using the command: `python3 -m pip install pandas` . The `numpy` is auto-installed through dependency management.

[2] Install using command: `python3 -m pip install docopt`

A Scripts

A.1 Script to convert formatted CSV file into a SACS seastate input file

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""Generate SACS storm load cards from a CSV file
slc.py 2022 ckunte

Tested for python v3.8.10, v3.10.8 with pandas >= v1.5.1

Usage: slc.py (-f <file>)
       slc.py --help
       slc.py --version

Options:
  -h, --help   Show this help
  -f --file    Specify CSV input file (required)

"""
import pandas as pd
from docopt import docopt


def main(*args):
    print("# Reading " + datfile + " file...", end="")
    df = pd.read_csv("./" + datfile)
    print("done.")
    print("FILE B")
    for i in range(len(df)):
        # PRINTING WAVE INPUT LINES
        print(f"LOADCN{i+1:4}")
        print(f"LOADLB{i+1:4}Envir for pile storm analysis")
        print(W[0])
        print(
            f"{W[0]:4}"  # col 1-4, line label
            + f"{W[1]:4}"  # col 5-8, kinematics fac.
            + f"{W[2]:4}"  # col 9-12, wave type
            + f"{df.iat[i, 0]:>6}"  # col 13-18, wave height
            + f"{F[0]:>6}"  # col 19-24, SWL, skip (from LDOPT)
            + f"{df.iat[i, 1]:>6}"  # col 25-30, wave period
            + f"{F[0]:>8}"  # col 31-38, wave length, skip if period is given
            + f"{df.iat[i, 2]:>6}"  # col 39-44, wave angle
            + f"{F[0]:>6}"  # col 45-50, mud line elev., skip (from LDOPT)
            + f"{W[3]:>0}"  # col 51, input mode
            + f"{W[4]:>7}"  # col 52-58, crest position
            + f"{W[5]:>6}"  # col 59-64, step size
            + f"{F[0]:1}"  # col 65-66, steps for dyn. analysis, skip
            + f"{W[6]:1}"  # col 67-68, static steps
            + f"{W[7]:1}"  # col 69-70, critical position
            + f"{W[8]:1}"  # col 71-72, member seg. (max)
            + f"{W[9]:1}"  # col 73-74, member seg. (min)
            # + "{0:0}".format(F[0])  # col 75, local accel. only, skip
            # + "{0:0}".format(F[0])  # col 76, print opt, skip
            # + "{0:<1}".format(F[0])  # col 77-78, order of stream func., skip
        )
        # PRINTING CURRENT INPUT LINES
        print(C[0])
        print(
            f"{C[0]:4}"  # col 1-4, line label
            + f"{F[0]:>4}"  # col 5-8, min inline curr velocity, skip
            + f"{eam[9]:>8}"  # col 9-16, elev above mud line
            + f"{df.iat[i, 12]:>8}"  # col 17-24, curr velocity
            + f"{df.iat[i, 2]:>8}"  # col 25-32, curr dir
            + f"{F[0]:>8}"  # col 33-40, mudline elev override, skip
            + f"{F[0]:>8}"  # col 41-48, blocking factor, skip
            + f"{F[0]:>8}"  # col 49-56, elev, skip
            + f"{C[1]:1}"  # col 57-58, elev, generate blocking fac.
            + f"{F[0]:>0}"  # col 59, null
            + f"{C[2]:1}"  # col 60-61, crest stretching opt.
            + f"{F[0]:>0}"  # col 62, null
            + f"{F[0]:2}"  # col 63-65, velocity units opt., skip
            + f"{F[0]:>0}"  # col 66, null
            + f"{F[0]:2}"  # col 67-69, elev percent opt., skip
            + f"{F[0]:>3}"  # col 70, null (for now this is a workaround)
            + f"{C[3]:>2}"  # col 71-73, AWP opt.
        )
        # adjust ranges depending upon the current profile
        for n, m in zip(range(8, -1, -1), range(11, 2, -1)):
            print(
                f"{C[0]}"  # col 1-4, line label
                + f"{F[0]:>4}"  # col 5-8, min inline curr velocity, skip
                + f"{eam[n]:>8}"  # col 9-16, elev above mud line
                + f"{df.iat[i, m]:>8}"  # col 17-24, curr velocity
                + f"{df.iat[i, 2]:>8}"  # col 25-32, curr dir
            )
    pass


if __name__ == "__main__":
    args = docopt(
        __doc__, version="Generate SACS storm load cards from a CSV file, v0.1"
    )
    datfile = "%s" % (args["<file>"])
    #
    # -- BEGIN USER INPUTS --
    #
    # WAVE DEFINITION AND POSITION PARAMETERS (SACS SEASTATE MANUAL, PG 170)
    #
    W = [
        "WAVE",  # line label
        0.95,  # kinematics factor
        "STOK",  # wave type
        "D",  # input mode (length (L), degree (D), or time (T))
        -90.0,  # crest position -- wave
        4.00,  # step size -- wave
        " 90",  # static steps -- wave
        "MM",  # critical position -- wave
        "10",  # member segmentation (max)
        " 1",  # member segmentation (min)
    ]
    # CURRENT PARAMETERS (SACS SEASTATE MANUAL, PG 171)
    #
    C = [
        "CURR",  # line label
        "BC",  # option to generate blocking factor
        "NL",  # crest stretching option
        "AWP",  # apparent wave period option
    ]
    # ELEVATION ABOVE MUDLINE (FOR CURRENT PROFILE)
    #
    eam = [
        166.18,
        151.18,
        141.18,
        121.18,
        101.18,
        81.18,
        61.18,
        41.18,
        21.18,
        1.18,
    ]
    # FILLER FOR EMPTY (OR NULL) COLUMN BLOCKS
    #
    F = [" "]
    #
    # CSV DATA FILE FROM METOCEAN TO USE
    #
    # Headers in CSV file:
    # H (m), T(s), ThetaP PlatformNth(Deg), WS (m/s), CS5(m/s), CS20(m/s), CS30(m/s), CS50(m/s), CS70(m/s), CS90(m/s), CS110(m/s), CS130(m/s), CS150(m/s), CS170(m/s)

    # -- END USER INPUTS --

    main(datfile, W, F, C)
```

A.2 Script for formatting csv file

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""Format CSV file with Pandas
fdf.py 2022 ckunte

Usage: fdf.py (-f <file>)
       fdf.py --help
       fdf.py --version

Options:
  -h, --help   Show this help
  -f --file    Specify CSV input file to format (required)

"""
import numpy as np
import pandas as pd
from docopt import docopt


def main(*args):
    print("# Reading " + datfile + " file...", end="")
    df = pd.read_csv("./" + datfile)
    print("done.")
    # remove wind speed column from data (by index -- this is a workaround:
    # [should be [3], but somehow [2] works -- possibly a python 3.8.10 bug)
    df2 = df.drop(df.columns[[2]], axis=1)
    return df2.to_csv("F" + datfile)


if __name__ == "__main__":
    args = docopt(
        __doc__, version="Generate SACS storm load cards from a CSV file, v0.1"
    )
    datfile = "%s" % (args["<file>"])
    main(datfile)
    print("Formatted file:", "F" + datfile)
```

B Metocean data file example

```
H  (m), T(s), ThetaP PlatformNth(Deg), WS  (m/s), CS5(m/s),  CS20(m/s),  CS30(m/s),  CS50(m/s),  CS70(m/s),  CS90(m/s), CS110(m/s), CS130(m/s), CS150(m/s), CS170(m/s)
  4.48,  14.56,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  4.81,  14.67,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  4.40,  14.21,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  4.18,  12.34,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  2.83,   8.32,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  3.76,  14.89,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  6.07,  14.04,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  4.13,  15.90,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  4.76,  13.01,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  3.94,  13.43,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  2.26,   8.39,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  1.66,  12.44,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  6.48,  11.75,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  6.07,  12.40,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  3.64,   9.72,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  4.74,   8.99,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  7.18,  12.71,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  5.18,  15.47,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  4.33,  14.93,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  2.18,  14.52,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  3.64,  13.86,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  4.83,  12.67,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  3.14,  11.35,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
  0.78,   5.32,290.00,  13.84,  0.75,   0.68,  0.60,   0.50,  0.44,   0.41,  0.35,   0.30,  0.25,   0.15,
```

...

C Sample output

```
# Reading FTS001.000040TS.csv file...done.
FILE B
LOADCN   1
LOADLB   1Envir for pile storm analysis
WAVE
WAVE0.95STOK  4.48       14.56         290.0      D  -90.0   4.0  90MM10 1
CURR
CURR        1.18    0.15   290.0                        BC NL         AWP
CURR       21.18    0.25   290.0
CURR       41.18     0.3   290.0
CURR       61.18    0.35   290.0
CURR       81.18    0.41   290.0
CURR      101.18    0.44   290.0
CURR      121.18     0.5   290.0
CURR      141.18     0.6   290.0
CURR      151.18    0.68   290.0
CURR      166.18    0.75   290.0
LOADCN   2
LOADLB   2Envir for pile storm analysis
WAVE
WAVE0.95STOK  4.81       14.67         290.0      D  -90.0   4.0  90MM10 1
CURR
CURR        1.18    0.15   290.0                        BC NL         AWP
CURR       21.18    0.25   290.0
CURR       41.18     0.3   290.0
CURR       61.18    0.35   290.0
CURR       81.18    0.41   290.0
CURR      101.18    0.44   290.0
CURR      121.18     0.5   290.0
CURR      141.18     0.6   290.0
CURR      151.18    0.68   290.0
CURR      166.18    0.75   290.0
```

...

```
CURR      166.18    0.83   185.0
LOADCN7930
LOADLB7930Envir for pile storm analysis
WAVE
WAVE0.95STOK  5.14       12.48         185.0      D  -90.0   4.0  90MM10 1
CURR
CURR        1.18    0.17   185.0                        BC NL         AWP
CURR       21.18    0.28   185.0
CURR       41.18    0.33   185.0
CURR       61.18    0.39   185.0
CURR       81.18    0.45   185.0
CURR      101.18    0.48   185.0
CURR      121.18    0.55   185.0
CURR      141.18    0.67   185.0
CURR      151.18    0.75   185.0
CURR      166.18    0.83   185.0
LOADCN7931
LOADLB7931Envir for pile storm analysis
WAVE
WAVE0.95STOK  6.63       11.36         185.0      D  -90.0   4.0  90MM10 1
CURR
CURR        1.18    0.17   185.0                        BC NL         AWP
CURR       21.18    0.28   185.0
CURR       41.18    0.33   185.0
CURR       61.18    0.39   185.0
CURR       81.18    0.45   185.0
CURR      101.18    0.48   185.0
CURR      121.18    0.55   185.0
CURR      141.18    0.67   185.0
CURR      151.18    0.75   185.0
CURR      166.18    0.83   185.0
```