```python
# -*- coding: utf-8 -*-
"""
Created on Thu Sep  5 13:23:30 2019

@author: VACALDER
"""

# PROGRAM TO ANALYZE DATA FROM BATCH RUN of NLTHA FOR TDPBEE
#   Victor A Calderon
#   PhD Student/ Research Assistant
#   NC STATE UNIVERSITY
#   2021 (c)
#
#
# ------------------------------------------------------------
# |                                IMPORTS
# ------------------------------------------------------------

import time

start_time = time.time()
import numpy as np
import pandas as pd
from LibUnitsMUS import *


# ------------------------------------------------------------

def Postprocessor_of_data(GM_fn, CL, CLt, D, SF, ALR, rhol, rhov):
    # 1. Opening folder to access data

    SpectrumDir = r'/home/vacalderon/Documents/MainshocksParallel_2.0.3/ResponseSpectrumAnalysis'
    rootdir = r'/home/vacalderon/Documents/MainshocksParallel_2.0.3/data'

    Es = 29000
    earthquake = []
```

```
37      PGA_MS = []
38      covers = []
39      times = []
40      WaterCement_Ratios = []
41      CorrosionLvls_Long = []
42      CorrosionLvls_Trans = []
43      Steel_Strains = []
44      CConc_Strains = []
45      UConc_Strains = []
46      YieldStresses = []
47      YielStressesTrans = []
48      AreaOfSteels = []
49      spacings = []
50      CoreDiameters = []
51      AxialLoads = []
52      Diameters = []
53      AreaRebars = []
54      BarDiameters = []
55      CompStrength = []
56      LS_ConcCover = []
57      LS_SteelBB = []
58      LS_ConfYield = []
59      FirstPeriods = []
60      EffectivePeriods = []
61      Forces = []
62      Displacements = []
63      SpectralDisplacement_Results = []
64      PGD_Results = []
65      Rho_ls = []
66      Rho_vs = []
67      Heights = []
68      AxialLoadRatios = []
69      SpectralDisplacement_Teff_xi = []
70      LSs = []
71      LSc = []
72      DCs = []
73      DCc = []
74      Us = []
75      Ductilities = []
76      datadir = rootdir + '/' + GM_fn + "/CL" + str(CL) + "/
   CLt" + str(CLt) + "/D" + str(D) + "/SF" + str(SF) + "/ALR"
    + str(
77          ALR) + "/RhoL" + str(rhol) + "/Rhov" + str(rhov)
```

```
 78
 79      # 2. Read Conditions
 80      groundmotion = GM_fn
 81      with open(datadir + "/PGA.out") as pgafile:
 82          linespgafile = pgafile.readline()
 83      pga = float(linespgafile.split()[0])
 84      with open(datadir + "/Conditions.out") as  conditions:
 85          linesconditions = conditions.readline()
 86
 87      CLl = float(linesconditions.split()[0])
 88
 89      # 3. Read Period of the Structure
 90      with open(datadir + "/Period.out") as  Period_01:
 91          lines_Period_01 = Period_01.readline()
 92      T1 = float(lines_Period_01.split()[0])
 93
 94      # 4. Read Material Properties for run
 95
 96      with open(datadir + "/mat.out") as material_prop:
 97          lines_material_prop = material_prop.readline()
 98
 99      YieldStress_Long = float(lines_material_prop.split()[0
    ])
100      YieldStress_Trans = float(lines_material_prop.split()[
    1])
101      AreaOfSteel = float(lines_material_prop.split()[2])
102      spacing_of_steel = float(lines_material_prop.split()[3
    ])
103      CoreDiameter = float(lines_material_prop.split()[4])
104      AxialLoad = float(lines_material_prop.split()[5])
105      Diameter = float(lines_material_prop.split()[6])
106      Height = float(lines_material_prop.split()[7])
107      AreaRebar = float(lines_material_prop.split()[8])
108      CompStrengths = float(lines_material_prop.split()[9])
109      AxialLoadRatio = float(lines_material_prop.split()[14
    ])
110      dbl = float(lines_material_prop.split()[15])
111      ros = (4 * AreaOfSteel) / (CoreDiameter *
    spacing_of_steel)
112      Ag = 0.25 * math.pi * Diameter ** 2
113      e_ss = 0.015
114      e_ccc = 0.004
115      e_bb = 0.03 + 700 * ros * YieldStress_Trans / Es - 0.1
```

```
115  * AxialLoad / (CompStrengths * Ag)
116      e_csy = 0.009 - 0.3 * AreaRebar / Ag + 3.9 *
    YieldStress_Trans / Es
117
118      e_cbs = 0.14-0.0045*CL
119      e_bb_barcley = np.log(e_cbs/0.001)/(300*ALR+0.7/rhov)
120
121      # 5. Force Displacement Plot
122
123      with open(datadir + "/DFree.out") as d:
124          linesd = d.readlines()
125      with open(datadir + "/RBase.out") as F:
126          linesf = F.readlines()
127
128      x = [line.split()[1] for line in linesd[:-1]]
129      y = [line.split()[1] for line in linesf[:-1]]
130
131      X = [float(i) for i in x]
132      Y = [-float(i) for i in y]
133      maxDisp = max(X)
134      minDisp = min(X)
135      if maxDisp > abs(minDisp):
136          AbsMaxDisp = maxDisp
137      elif maxDisp < abs(minDisp):
138          AbsMaxDisp = minDisp
139      maxDispPoss = X.index(AbsMaxDisp)
140      maxForce_at_maxDisp = Y[maxDispPoss]
141      Keff = abs(maxForce_at_maxDisp) / abs(AbsMaxDisp)
142      meff = AxialLoad * kip / g
143      Teff = (2 * math.pi) * (math.sqrt((meff / Keff)))
144      Lsp = 0.15 * YieldStress_Long * dbl
145      e_steel_yield = YieldStress_Long/Es
146      phi_y = 2.25 * e_steel_yield/Diameter
147      delta_y = phi_y * (Height + Lsp) ** 2 / 3
148      delta_u = AbsMaxDisp
149      mu = abs(delta_u) / delta_y
150
151
152      # 6. Steel Stress Strain Analysis
153      with open(datadir + "/StressStrain.out") as
    SteelStressStrain1:
154          linesSteelStressStrain1 = SteelStressStrain1.
    readlines()
```

```
155     StlStress1 = [line.split()[1] for line in
    linesSteelStressStrain1]
156     StlStrain1 = [line.split()[-1] for line in
    linesSteelStressStrain1]
157     siGM_fnaStl1 = [float(i) for i in StlStress1]
158     epsilonStl1 = [float(i) for i in StlStrain1[:-1]]
159
160     with open(datadir + "/StressStrain4.out") as
    SteelStressStrain2:
161         linesSteelStressStrain2 = SteelStressStrain2.
    readlines()
162     StlStress2 = [line.split()[1] for line in
    linesSteelStressStrain2]
163     StlStrain2 = [line.split()[-1] for line in
    linesSteelStressStrain2]
164     siGM_fnaStl2 = [float(i) for i in StlStress2]
165     epsilonStl2 = [float(i) for i in StlStrain2[:-1]]
166
167     # 7. Confined Concrete Stress Strain Analysis
168     with open(datadir + "/StressStrain2.out") as
    CConcStressStrain:
169         linesCConcStressStrain = CConcStressStrain.
    readlines()
170     CConcStress = [line.split()[1] for line in
    linesCConcStressStrain]
171     CConcStrain = [line.split()[2] for line in
    linesCConcStressStrain]
172     siGM_fnaCConc = [float(i) for i in CConcStress]
173     epsilonCConc = [float(i) for i in CConcStrain[:-1]]
174
175     # 8. UncConfined Concrete Stress Strain Analysis
176     with open(datadir + "/StressStrain3.out") as
    UnConcStressStrain:
177         linesUnConcStressStrain = UnConcStressStrain.
    readlines()
178     UnConcStress = [line.split()[1] for line in
    linesUnConcStressStrain]
179     UnConcStrain = [line.split()[2] for line in
    linesUnConcStressStrain]
180     siGM_fnaUnConc = [float(i) for i in UnConcStress]
181     epsilonUnConc = [float(i) for i in UnConcStrain[:-1]]
182
183     # 9. Writing SD_teff
```

```
184     SpectrumFile = open(SpectrumDir + '/' + groundmotion
     + '.csv')
185     SpectrumContent = SpectrumFile.readlines()
186     SDC = SpectrumContent[12:109]
187     SDC_cols = ['Period', 'SD', 'PSV', 'PSA']
188     SDC_Data = [line.split(',') for line in SDC[:]]
189     SDC_DF = pd.DataFrame(columns=SDC_cols, data=SDC_Data)
190     PeriodStringList = list(SDC_DF['Period'])
191     SpectralDisplacementStringList = list(SDC_DF['SD'])
192     PGD = float(SpectralDisplacementStringList[-1])
193     T = [float(i) for i in PeriodStringList]
194     SpectralDisplacementList = list(SDC_DF['SD'])
195     SD_Float = [float(i) for i in SpectralDisplacementList
    ]
196     SD_at_Teff = np.interp(Teff, T, SD_Float)
197
198     if mu > 1:
199         xi_eq = 0.05 + 0.565 * (mu - 1) / (mu * np.pi)
200         DF = np.sqrt((0.07) / (0.05 + xi_eq))
201         SD_Teff_xi_eq = DF * SD_at_Teff
202
203     elif mu <= 1:
204         SD_Teff_xi_eq = SD_at_Teff
205
206
207     #10. Collapse analysis for strains
208     e_steel_max = max(max(max(epsilonStl1), max(
    epsilonStl2)), abs(min(min(epsilonStl1), min(epsilonStl2
    ))))
209     e_concrete_max = -min(epsilonCConc)
210     #10.1 Steel Serviciability
211     if e_steel_max < e_ss:
212         steel_serviciability = 0
213     elif e_steel_max > e_ss:
214         steel_serviciability = 1
215     #10.2 Concrete Serviciability
216     if e_concrete_max < e_ccc:
217         concrete_serviciability = 0
218     elif e_concrete_max > e_ccc:
219         concrete_serviciability = 1
220     #10.3 Concrete Damage Control
221     if e_concrete_max < e_csy:
222         concrete_damage = 0
```

```
223     elif e_concrete_max > e_csy:
224         concrete_damage = 1
225     #10.4 Steel Damage Control
226     if e_steel_max < e_bb:
227         steel_damage = 0
228     elif e_steel_max > e_bb:
229         steel_damage = 1
230     #10.5 Steel Ultimate (Barcley)
231     if e_steel_max < e_bb_barcley:
232         steel_ultimate = 0
233     elif e_steel_max > e_bb_barcley:
234         steel_ultimate = 1
235 
236 # 11. Writing data to variables
237 
238     earthquake.append(groundmotion)
239     PGA_MS.append(pga)
240     CorrosionLvls_Long.append(CLl)
241     CorrosionLvls_Trans.append(CLt)
242     Steel_Strains.append(max(max(max(epsilonStl1), max(
    epsilonStl2)), abs(min(min(epsilonStl1), min(epsilonStl2
    )))))
243     CConc_Strains.append(-min(epsilonCConc))
244     UConc_Strains.append(-min(epsilonUnConc))
245     YieldStresses.append(YieldStress_Long)
246     YielStressesTrans.append(YieldStress_Trans)
247     AreaOfSteels.append(AreaOfSteel)
248     spacings.append(spacing_of_steel)
249     CoreDiameters.append(CoreDiameter)
250     AxialLoads.append(AxialLoad)
251     Diameters.append(Diameter)
252     AreaRebars.append(AreaRebar)
253     BarDiameters.append(dbl)
254     CompStrength.append(-CompStrengths)
255     LS_ConcCover.append(e_ccc)
256     LS_ConfYield.append(e_csy)
257     LS_SteelBB.append(e_bb)
258     FirstPeriods.append(T1)
259     EffectivePeriods.append(Teff)
260     Forces.append(maxForce_at_maxDisp)
261     Displacements.append(AbsMaxDisp)
262     SpectralDisplacement_Results.append(SD_at_Teff)
263     PGD_Results.append(PGD)
```

```
264     Rho_ls.append(rhol)
265     Rho_vs.append(rhov)
266     Heights.append(Height)
267     AxialLoadRatios.append(AxialLoadRatio)
268     LSs.append(steel_serviciability)
269     LSc.append(concrete_serviciability)
270     DCs.append(steel_damage)
271     DCc.append(concrete_damage)
272     Us.append(steel_ultimate)
273     SpectralDisplacement_Teff_xi.append(SD_Teff_xi_eq)
274     Ductilities.append(mu)
275 
276     # 10. Preparing dictionary to wirte output database
277 
278     dataDict = {'earthquake': earthquake,
279                 'pga_(g)': PGA_MS,
280                 'CorrosionLvl_Long': CorrosionLvls_Long,
281                 'CorrosionLvl_Trans': CorrosionLvls_Trans,
282                 'First_Period_s': FirstPeriods,
283                 'Steel_Strain': Steel_Strains,
284                 'Conf_Conc_Strain': CConc_Strains,
285                 'Unc_Conc_srain': UConc_Strains,
286                 'Fy_ksi': YieldStresses,
287                 'fyt_ksi': YielStressesTrans,
288                 'Ast_in2': AreaOfSteels,
289                 'st_in': spacings,
290                 'Dprime_in': CoreDiameters,
291                 'PCol_kip': AxialLoads,
292                 'DCol_in': Diameters,
293                 'barAreaSec_in2': AreaRebars,
294                 'fc_ksi': CompStrength,
295                 'LimitState_ConcreteCoverCrushing':
    LS_ConcCover,
296                 'ConfinementSteelYielding': LS_ConfYield,
297                 'LongitudinalSteelBuckling': LS_SteelBB,
298                 'Effective period, Teff': EffectivePeriods
    ,
299                 'Force': Forces,
300                 'MaxDisplacement at MaxForce':
    Displacements,
301                 'SD_at_Teff': SpectralDisplacement_Results
    ,
302                 'SD_Teff_xi':SpectralDisplacement_Teff_xi,
```

```
303                     'rhol': Rho_ls,
304                     'rhov': Rho_vs,
305                     'ALR': AxialLoadRatios,
306                     'height_of_col': Heights,
307                     'long_bar_diameter':BarDiameters,
308                     'ServciabilitySteel': LSs,
309                     'ServiciabilityConcrete': LSc,
310                     'DamageControlSteel': DCs,
311                     'DamageControlConcrete': DCc,
312                     'UltimateSteel': Us,
313                     'Ductility': Ductilities}
314
315     # 11. Generating data frame to write data to csv file
316
317     DataFrame_Out = pd.DataFrame(dataDict)
318
319     # 12. Writing CSV File
320     DataFrame_Out.to_csv('/home/vacalderon/Documents/
    MainshocksParallel_2.0.3/results/PosprocData.csv', mode='a
    ',
321                          header=False)
322
323     # Output to show in console
324     print(
    '-------------------------------------------------------
    ------------------------------------------------')
325     print("POSTPROCESSING COMPLETE")
326     print(
    '-------------------------------------------------------
    ------------------------------------------------')
327     print("--- %s minutes ---" % ((time.time() -
    start_time) / 60))
328
```