```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 = []
    PGA_MS = []
    covers = []
    times = []
    WaterCement_Ratios = []
    CorrosionLvls_Long = []
    CorrosionLvls_Trans = []
    Steel_Strains = []
    CConc_Strains = []
    UConc_Strains = []
    YieldStresses = []
    YielStressesTrans = []
    AreaOfSteels = []
    spacings = []
    CoreDiameters = []
    AxialLoads = []
    Diameters = []
    AreaRebars = []
    BarDiameters = []
    CompStrength = []
    LS_ConcCover = []
```

```
LS_SteelBB = []
LS_ConfYield = []
FirstPeriods = []
EffectivePeriods = []
Forces = []
Displacements = []
SpectralDisplacement_Results = []
PGD_Results = []
Rho_ls = []
Rho_vs = []
Heights = []
AxialLoadRatios = []
SpectralDisplacement_Teff_xi = []
LSs = []
LSc = []
DCs = []
DCc = []
Us = []
Ductilities = []
datadir = rootdir + '/' + GM_fn + "/CL" + str(CL) + "/CLt" + str(CLt) + "/D" + str(D) + "/SF" +
    ALR) + "/RhoL" + str(rhol) + "/Rhov" + str(rhov)

# 2. Read Conditions
groundmotion = GM_fn
with open(datadir + "/PGA.out") as pgafile:
    linespgafile = pgafile.readline()
pga = float(linespgafile.split()[0])
with open(datadir + "/Conditions.out") as  conditions:
    linesconditions = conditions.readline()

CLl = float(linesconditions.split()[0])

# 3. Read Period of the Structure
with open(datadir + "/Period.out") as  Period_01:
    lines_Period_01 = Period_01.readline()
T1 = float(lines_Period_01.split()[0])

# 4. Read Material Properties for run

with open(datadir + "/mat.out") as material_prop:
    lines_material_prop = material_prop.readline()

YieldStress_Long = float(lines_material_prop.split()[0])
YieldStress_Trans = float(lines_material_prop.split()[1])
AreaOfSteel = float(lines_material_prop.split()[2])
spacing_of_steel = float(lines_material_prop.split()[3])
CoreDiameter = float(lines_material_prop.split()[4])
AxialLoad = float(lines_material_prop.split()[5])
Diameter = float(lines_material_prop.split()[6])
Height = float(lines_material_prop.split()[7])
AreaRebar = float(lines_material_prop.split()[8])
CompStrengths = float(lines_material_prop.split()[9])
AxialLoadRatio = float(lines_material_prop.split()[14])
dbl = float(lines_material_prop.split()[15])
ros = (4 * AreaOfSteel) / (CoreDiameter * spacing_of_steel)
Ag = 0.25 * math.pi * Diameter ** 2
```

```
e_ss = 0.015
e_ccc = 0.004
e_bb = 0.03 + 700 * ros * YieldStress_Trans / Es - 0.1 * AxialLoad / (CompStrengths * Ag)
e_csy = 0.009 - 0.3 * AreaRebar / Ag + 3.9 * YieldStress_Trans / Es

e_cbs = 0.14-0.0045*CL
e_bb_barcley = np.log(e_cbs/0.001)/(300*ALR+0.7/rhov)

# 5. Force Displacement Plot

with open(datadir + "/DFree.out") as d:
    linesd = d.readlines()
with open(datadir + "/RBase.out") as F:
    linesf = F.readlines()

x = [line.split()[1] for line in linesd[:-1]]
y = [line.split()[1] for line in linesf[:-1]]

X = [float(i) for i in x]
Y = [-float(i) for i in y]
maxDisp = max(X)
minDisp = min(X)
if maxDisp > abs(minDisp):
    AbsMaxDisp = maxDisp
elif maxDisp < abs(minDisp):
    AbsMaxDisp = minDisp
maxDispPoss = X.index(AbsMaxDisp)
maxForce_at_maxDisp = Y[maxDispPoss]
Keff = abs(maxForce_at_maxDisp) / abs(AbsMaxDisp)
meff = AxialLoad * kip / g
Teff = (2 * math.pi) * (math.sqrt((meff / Keff)))
Lsp = 0.15 * YieldStress_Long * dbl
e_steel_yield = YieldStress_Long/Es
phi_y = 2.25 * e_steel_yield/Diameter
delta_y = phi_y * (Height + Lsp) ** 2 / 3
delta_u = AbsMaxDisp
mu = abs(delta_u) / delta_y


# 6. Steel Stress Strain Analysis
with open(datadir + "/StressStrain.out") as SteelStressStrain1:
    linesSteelStressStrain1 = SteelStressStrain1.readlines()
StlStress1 = [line.split()[1] for line in linesSteelStressStrain1]
StlStrain1 = [line.split()[-1] for line in linesSteelStressStrain1]
siGM_fnaStl1 = [float(i) for i in StlStress1]
epsilonStl1 = [float(i) for i in StlStrain1[:-1]]

with open(datadir + "/StressStrain4.out") as SteelStressStrain2:
    linesSteelStressStrain2 = SteelStressStrain2.readlines()
StlStress2 = [line.split()[1] for line in linesSteelStressStrain2]
StlStrain2 = [line.split()[-1] for line in linesSteelStressStrain2]
siGM_fnaStl2 = [float(i) for i in StlStress2]
epsilonStl2 = [float(i) for i in StlStrain2[:-1]]

# 7. Confined Concrete Stress Strain Analysis
with open(datadir + "/StressStrain2.out") as CConcStressStrain:
    linesCConcStressStrain = CConcStressStrain.readlines()
CConcStress = [line.split()[1] for line in linesCConcStressStrain]
CConcStrain = [line.split()[2] for line in linesCConcStressStrain]
siGM_fnaCConc = [float(i) for i in CConcStress]
epsilonCConc = [float(i) for i in CConcStrain[:-1]]

# 8. UncConfined Concrete Stress Strain Analysis
with open(datadir + "/StressStrain3.out") as UnConcStressStrain:
    linesUnConcStressStrain = UnConcStressStrain.readlines()
UnConcStress = [line.split()[1] for line in linesUnConcStressStrain]
UnConcStrain = [line.split()[2] for line in linesUnConcStressStrain]
siGM_fnaUnConc = [float(i) for i in UnConcStress]
epsilonUnConc = [float(i) for i in UnConcStrain[:-1]]

# 9. Writing SD_teff
SpectrumFile = open(SpectrumDir + '/' + groundmotion + '.csv')
SpectrumContent = SpectrumFile.readlines()
SDC = SpectrumContent[12:109]
SDC_cols = ['Period', 'SD', 'PSV', 'PSA']
SDC_Data = [line.split(',') for line in SDC[:]]
SDC_DF = pd.DataFrame(columns=SDC_cols, data=SDC_Data)
PeriodStringList = list(SDC_DF['Period'])
SpectralDisplacementStringList = list(SDC_DF['SD'])
PGD = float(SpectralDisplacementStringList[-1])
T = [float(i) for i in PeriodStringList]
SpectralDisplacementList = list(SDC_DF['SD'])
SD_Float = [float(i) for i in SpectralDisplacementList]
SD_at_Teff = np.interp(Teff, T, SD_Float)

if mu > 1:
    xi_eq = 0.05 + 0.565 * (mu - 1) / (mu * np.pi)
    DF = np.sqrt((0.07) / (0.05 + xi_eq))
    SD_Teff_xi_eq = DF * SD_at_Teff

elif mu <= 1:
    SD_Teff_xi_eq = SD_at_Teff


#10. Collapse analysis for strains
e_steel_max = max(max(max(epsilonStl1), max(epsilonStl2)), abs(min(min(epsilonStl1), min(epsilo
e_concrete_max = -min(epsilonCConc)
#10.1 Steel Serviciability
if e_steel_max < e_ss:
    steel_serviciability = 0
elif e_steel_max > e_ss:
    steel_serviciability = 1
#10.2 Concrete Serviciability
if e_concrete_max < e_ccc:
    concrete_serviciability = 0
elif e_concrete_max > e_ccc:
    concrete_serviciability = 1
#10.3 Concrete Damage Control
if e_concrete_max < e_csy:
    concrete_damage = 0
elif e_concrete_max > e_csy:
    concrete_damage = 1
    #10.4 Steel Damage Control
    if e_steel_max < e_bb:
        steel_damage = 0
    elif e_steel_max > e_bb:
        steel_damage = 1
    #10.5 Steel Ultimate (Barcley)
    if e_steel_max < e_bb_barcley:
        steel_ultimate = 0
    elif e_steel_max > e_bb_barcley:
        steel_ultimate = 1

# 11. Writing data to variables

    earthquake.append(groundmotion)
    PGA_MS.append(pga)
    CorrosionLvls_Long.append(CLl)
    CorrosionLvls_Trans.append(CLt)
    Steel_Strains.append(max(max(max(epsilonStl1), max(epsilonStl2)), abs(min(min(epsilonStl1), mi
    CConc_Strains.append(-min(epsilonCConc))
    UConc_Strains.append(-min(epsilonUnConc))
    YieldStresses.append(YieldStress_Long)
    YielStressesTrans.append(YieldStress_Trans)
    AreaOfSteels.append(AreaOfSteel)
    spacings.append(spacing_of_steel)
    CoreDiameters.append(CoreDiameter)
    AxialLoads.append(AxialLoad)
    Diameters.append(Diameter)
    AreaRebars.append(AreaRebar)
    BarDiameters.append(dbl)
    CompStrength.append(-CompStrengths)
    LS_ConcCover.append(e_ccc)
    LS_ConfYield.append(e_csy)
    LS_SteelBB.append(e_bb)
    FirstPeriods.append(T1)
    EffectivePeriods.append(Teff)
    Forces.append(maxForce_at_maxDisp)
    Displacements.append(AbsMaxDisp)
    SpectralDisplacement_Results.append(SD_at_Teff)
    PGD_Results.append(PGD)
    Rho_ls.append(rhol)
    Rho_vs.append(rhov)
    Heights.append(Height)
    AxialLoadRatios.append(AxialLoadRatio)
    LSs.append(steel_serviciability)
    LSc.append(concrete_serviciability)
    DCs.append(steel_damage)
    DCc.append(concrete_damage)
    Us.append(steel_ultimate)
    SpectralDisplacement_Teff_xi.append(SD_Teff_xi_eq)
    Ductilities.append(mu)

    # 10. Preparing dictionary to wirte output database

    dataDict = {'earthquake': earthquake,
                'pga_(g)': PGA_MS,
                'CorrosionLvl_Long': CorrosionLvls_Long,
            'CorrosionLvl_Trans': CorrosionLvls_Trans,
            'First_Period_s': FirstPeriods,
            'Steel_Strain': Steel_Strains,
            'Conf_Conc_Strain': CConc_Strains,
            'Unc_Conc_srain': UConc_Strains,
            'Fy_ksi': YieldStresses,
            'fyt_ksi': YielStressesTrans,
            'Ast_in2': AreaOfSteels,
            'st_in': spacings,
            'Dprime_in': CoreDiameters,
            'PCol_kip': AxialLoads,
            'DCol_in': Diameters,
            'barAreaSec_in2': AreaRebars,
            'fc_ksi': CompStrength,
            'LimitState_ConcreteCoverCrushing': LS_ConcCover,
            'ConfinementSteelYielding': LS_ConfYield,
            'LongitudinalSteelBuckling': LS_SteelBB,
            'Effective period, Teff': EffectivePeriods,
            'Force': Forces,
            'MaxDisplacement at MaxForce': Displacements,
            'SD_at_Teff': SpectralDisplacement_Results,
            'SD_Teff_xi':SpectralDisplacement_Teff_xi,
            'rhol': Rho_ls,
            'rhov': Rho_vs,
            'ALR': AxialLoadRatios,
            'height_of_col': Heights,
            'long_bar_diameter':BarDiameters,
            'ServciabilitySteel': LSs,
            'ServiciabilityConcrete': LSc,
            'DamageControlSteel': DCs,
            'DamageControlConcrete': DCc,
            'UltimateSteel': Us,
            'Ductility': Ductilities}

# 11. Generating data frame to write data to csv file

DataFrame_Out = pd.DataFrame(dataDict)

# 12. Writing CSV File
DataFrame_Out.to_csv('/home/vacalderon/Documents/MainshocksParallel_2.0.3/results/PosprocData.c
                     header=False)

# Output to show in console
print('---------------------------------------------------------------------------------------
print("POSTPROCESSING COMPLETE")
print('---------------------------------------------------------------------------------------
print("--- %s minutes ---" % ((time.time() - start_time) / 60))
```