

Import libraries

```
In [1]: import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

```
In [122...]:
# CF_0_45_4_20cst_AT
CF_0_45_4_20cst_AT_001 = pd.read_excel('CF_0_45_4_20cst_AT_001.xlsx')
CF_0_45_4_20cst_AT_002 = pd.read_excel('CF_0_45_4_20cst_AT_002.xlsx')
CF_0_45_4_20cst_AT_003 = pd.read_excel('CF_0_45_4_20cst_AT_003.xlsx')
CF_0_45_4_20cst_AT_004 = pd.read_excel('CF_0_45_4_20cst_AT_004.xlsx')

# CF_0_45_4_100cst_AT
CF_0_45_4_100cst_AT_001 = pd.read_excel('CF_0_45_4_100cst_AT_001.xlsx')
CF_0_45_4_100cst_AT_002 = pd.read_excel('CF_0_45_4_100cst_AT_002.xlsx')
CF_0_45_4_100cst_AT_003 = pd.read_excel('CF_0_45_4_100cst_AT_003.xlsx')
CF_0_45_4_100cst_AT_004 = pd.read_excel('CF_0_45_4_100cst_AT_004.xlsx')

# CF_0_45_4_Dry_AT
CF_0_45_4_Dry_AT_001 = pd.read_excel('CF_0_45_4_Dry_AT_001.xlsx')
CF_0_45_4_Dry_AT_002 = pd.read_excel('CF_0_45_4_Dry_AT_002.xlsx')
CF_0_45_4_Dry_AT_003 = pd.read_excel('CF_0_45_4_Dry_AT_003.xlsx')
CF_0_45_4_Dry_AT_004 = pd.read_excel('CF_0_45_4_Dry_AT_004.xlsx')

# CF_0_45_4_Dry_IT
CF_0_45_4_Dry_IT_001 = pd.read_excel('CF_0_45_4_Dry_IT_001.xlsx')
CF_0_45_4_Dry_IT_002 = pd.read_excel('CF_0_45_4_Dry_IT_002.xlsx')
CF_0_45_4_Dry_IT_003 = pd.read_excel('CF_0_45_4_Dry_IT_003.xlsx')
CF_0_45_4_Dry_IT_004 = pd.read_excel('CF_0_45_4_Dry_IT_004.xlsx')
```

In [123...]

```
# CF_08_20cst_AT
CF_08_20cst_AT_001 = pd.read_excel('CF_08_20cst_AT_001.xlsx')
CF_08_20cst_AT_002 = pd.read_excel('CF_08_20cst_AT_002.xlsx')
CF_08_20cst_AT_003 = pd.read_excel('CF_08_20cst_AT_003.xlsx')
CF_08_20cst_AT_004 = pd.read_excel('CF_08_20cst_AT_004.xlsx')

# CF_08_100cst_AT
CF_08_100cst_AT_001 = pd.read_excel('CF_08_100cst_AT_001.xlsx')
CF_08_100cst_AT_002 = pd.read_excel('CF_08_100cst_AT_002.xlsx')
CF_08_100cst_AT_003 = pd.read_excel('CF_08_100cst_AT_003.xlsx')
CF_08_100cst_AT_004 = pd.read_excel('CF_08_100cst_AT_004.xlsx')

# CF_08_Dry_AT
CF_08_Dry_AT_001 = pd.read_excel('CF_08_Dry_AT_001.xlsx')
CF_08_Dry_AT_002 = pd.read_excel('CF_08_Dry_AT_002.xlsx')
CF_08_Dry_AT_003 = pd.read_excel('CF_08_Dry_AT_003.xlsx')
CF_08_Dry_AT_004 = pd.read_excel('CF_08_Dry_AT_004.xlsx')

# CF_08_Dry_IT
CF_08_Dry_IT_001 = pd.read_excel('CF_08_Dry_IT_001.xlsx')
CF_08_Dry_IT_002 = pd.read_excel('CF_08_Dry_IT_002.xlsx')
CF_08_Dry_IT_003 = pd.read_excel('CF_08_Dry_IT_003.xlsx')
CF_08_Dry_IT_004 = pd.read_excel('CF_08_Dry_IT_004.xlsx')
```

In [149...]

```
def first_compaction(specimen1, specimen2, specimen3 = pd.DataFrame({'A' : []}), specimen4 = pd.DataFrame({'A' : []})):
    numRows = 10491

    vector_1 = np.zeros((1,specimen1['Weg_mm'].size), dtype=int)
    vector_1[0,0:10491] = 1
    vector_1 = np.ma.masked_equal(vector_1, 1)

    vector_2 = np.zeros((1,specimen2['Weg_mm'].size), dtype=int)
    vector_2[0,0:10491] = 1
    vector_2 = np.ma.masked_equal(vector_2, 1)

    if (not specimen3.empty) & (specimen4.empty):
        vector_3 = np.zeros((1,specimen3['Weg_mm'].size), dtype=int)
        vector_3[0,0:10491] = 1
        vector_3 = np.ma.masked_equal(vector_3, 1)

    numCols = 3

    elif (specimen3.empty) & (not specimen4.empty):
        vector_4 = np.zeros((1,specimen4['Weg_mm'].size), dtype=int)
        vector_4[0,0:10491] = 1
        vector_4 = np.ma.masked_equal(vector_4, 1)

    numCols = 3

    elif (not specimen3.empty) & (not specimen4.empty):
        vector_3 = np.zeros((1,specimen3['Weg_mm'].size), dtype=int)
        vector_3[0,0:10491] = 1
        vector_3 = np.ma.masked_equal(vector_3, 1)

        vector_4 = np.zeros((1,specimen4['Weg_mm'].size), dtype=int)
        vector_4[0,0:10491] = 1
        vector_4 = np.ma.masked_equal(vector_4, 1)
```

```

    numCols = 4
elif (specimen3.empty) & (specimen4.empty):
    numCols = 2

# Force dataframe
force = pd.DataFrame(index=range(numRows),columns=range(numCols))
force.iloc[:, 0] = specimen1['Kraft_N'][vector_1.mask[0]]
force.iloc[:, 1] = specimen2['Kraft_N'][vector_2.mask[0]]
if (not specimen3.empty):
    force.iloc[:, 2] = specimen3['Kraft_N'][vector_3.mask[0]]
if (not specimen4.empty):
    force.iloc[:, 3] = specimen4['Kraft_N'][vector_4.mask[0]]

## mean value and std
force_mean = np.mean(force, axis = 1)
force_std = np.std(force, axis = 1)
force_std_range = (force_mean - force_std, force_mean + force_std)

# Displacement dataframe
displacement = pd.DataFrame(index=range(numRows),columns=range(numCols))
displacement.iloc[:, 0] = specimen1['Weg_mm'][vector_1.mask[0]]
displacement.iloc[:, 1] = specimen2['Weg_mm'][vector_2.mask[0]]
initial_value = 0
if (not specimen3.empty):
    displacement.iloc[:, 2] = specimen3['Weg_mm'][vector_3.mask[0]]
if (not specimen4.empty):
    displacement.iloc[:, 3] = specimen4['Weg_mm'][vector_4.mask[0]]

## mean value and std
displacement_mean = np.absolute(initial_value - np.mean(displacement, axis = 1))
displacement_std = np.std(displacement, axis = 1)
displacement_std_range = (displacement_mean - displacement_std, displacement_mean)

dictionary = {'Force-mean': force_mean, 'Force-std': force_std_range, 'Displacement-mean': displacement_mean, 'Displacement-std': displacement_std_range}

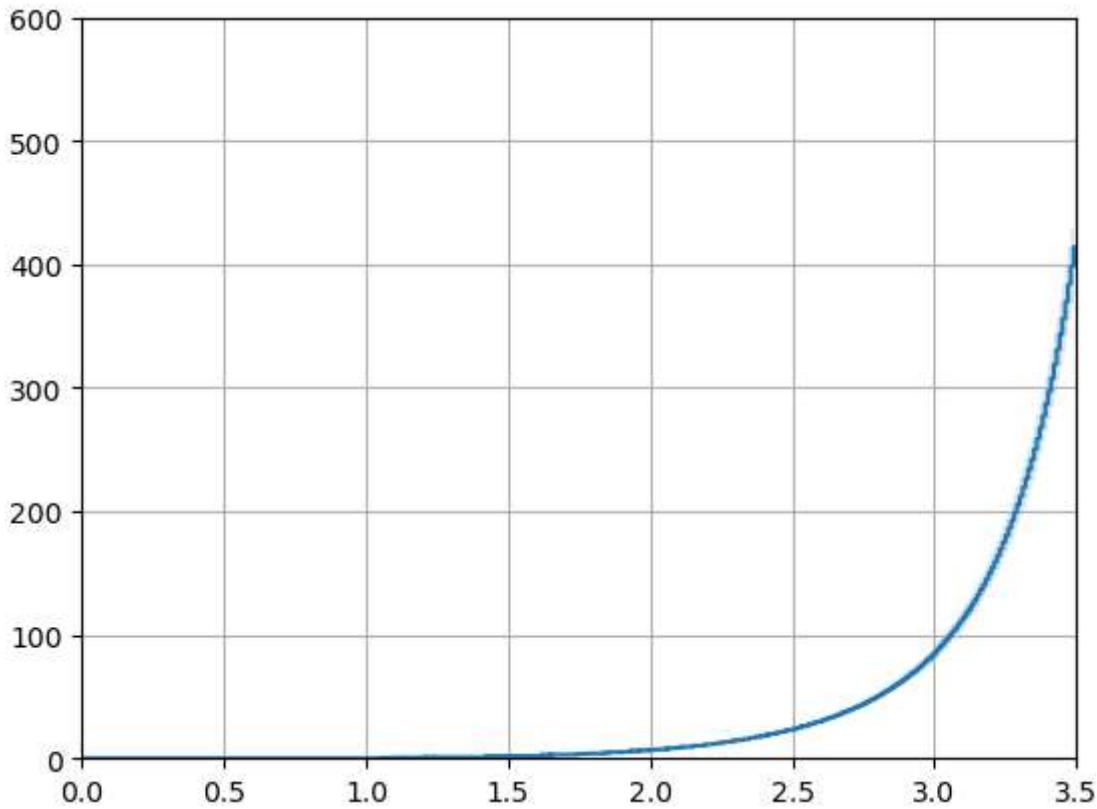
fig, ax1 = plt.subplots()
ax1.set_xlim(0,3.5)
ax1.set_ylim(0,600)
ax1.plot(displacement_mean, force_mean)
ax1.fill_between(dictionary['Displacement-mean'], dictionary['Force-std'][0], dictionary['Force-std'][1])
ax1.grid()

return dictionary

```

In [150...]

```
CF_0_45_4_Dry_AT = first_compaction(CF_0_45_4_Dry_AT_003, CF_0_45_4_Dry_AT_001, CF_0_45_4_Dry_AT_002)
```



```
In [4]: plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True

fig, ax1 = plt.subplots()

x = CF_08_20cst_AT_001.iloc[:,0]
y1 = CF_08_20cst_AT_001.iloc[:,1]
y2 = CF_08_20cst_AT_001.iloc[:,2]
y2 = 6 - y2

x2 = CF_08_20cst_AT_002.iloc[:,0]
y3 = CF_08_20cst_AT_002.iloc[:,1]
y4 = CF_08_20cst_AT_002.iloc[:,2]
y4 = 6 - y4

x3 = CF_08_20cst_AT_003.iloc[:,0]
y5 = CF_08_20cst_AT_003.iloc[:,1]
y6 = CF_08_20cst_AT_003.iloc[:,2]
y6 = 6 - y6

x4 = CF_08_20cst_AT_004.iloc[:,0]
y7 = CF_08_20cst_AT_004.iloc[:,1]
y8 = CF_08_20cst_AT_004.iloc[:,2]
y8 = 6 - y8

color = 'red'

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x, y1, color=color, marker = 's', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
```

```

ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x2, y3, color=color, marker = 'o', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x3, y5, color=color, marker = '^', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x4, y7, color=color, marker = '*', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'blue'
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x, y2, color=color, marker = 's', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

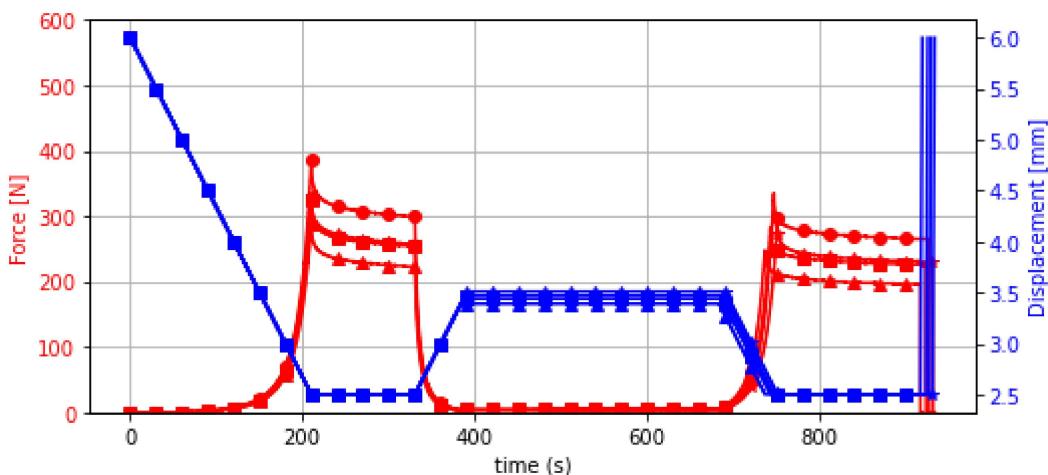
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x2, y4, color=color, marker = 'o', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x3, y6, color=color, marker = '^', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x4, y8, color=color, marker = '*', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

#plt.title('CF_08_20cst_AT')
ax1.grid()
ax1.set_ylim(0,600)
plt.show()
fig.savefig('CF_08_20cst_AT')

```



```

In [6]: plt.rcParams["figure.figsize"] = [15, 7]
plt.rcParams["figure.autolayout"] = True

fig, ax1 = plt.subplots()

```

```

x = CF_08_20cst_AT_001.iloc[0:46000,0]
y1 = CF_08_20cst_AT_001.iloc[0:46000,1]
y2 = CF_08_20cst_AT_001.iloc[0:46000,2]
y2 = 6 - y2

color = 'red'

ax1.set_xlabel('time (s)', fontsize = 30)
ax1.set_ylabel('Force [N]', fontsize = 30)
ax1.plot(x, y1, color=color, marker = 's', markevery = 1500, markersize = 10)
ax1.tick_params(axis='y')

ax2 = ax1.twinx()

color = 'blue'
ax2.set_ylabel('Displacement [mm]', fontsize = 30)
ax2.plot(x, y2, color=color, marker = 'o', markersize = 10, markevery = 1500)
ax2.tick_params(axis='y')

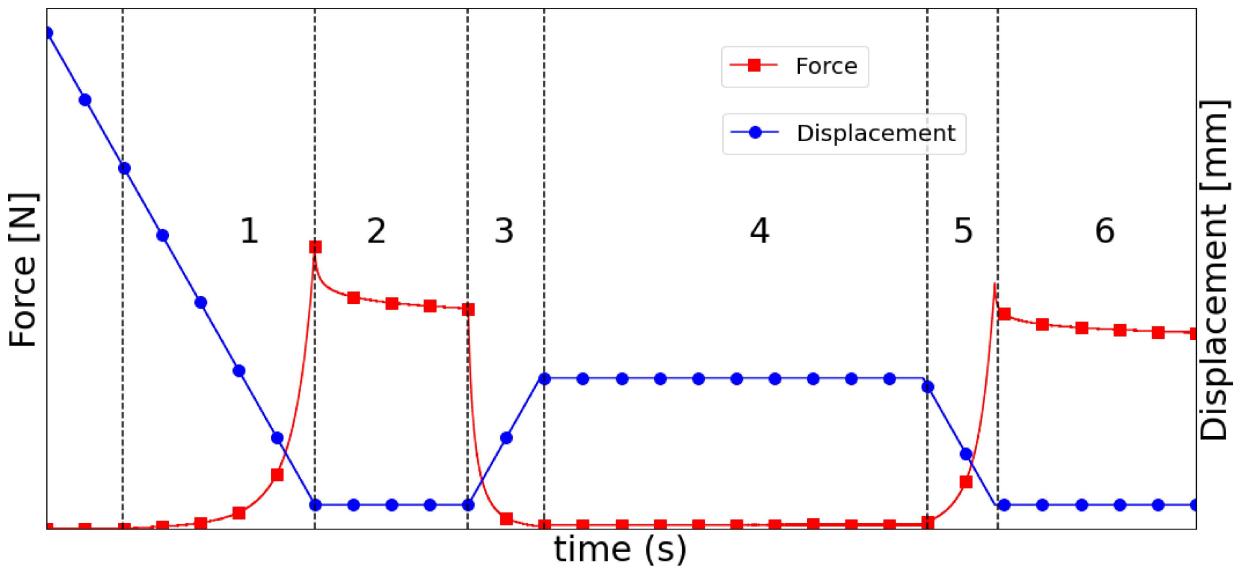
plt.axvline(x = 60, color = 'k', linestyle = 'dashed')
plt.axvline(x = 210, color = 'k', linestyle = 'dashed')
plt.axvline(x = 330, color = 'k', linestyle = 'dashed')
plt.axvline(x = 390, color = 'k', linestyle = 'dashed')
plt.axvline(x = 690, color = 'k', linestyle = 'dashed')
plt.axvline(x = 745, color = 'k', linestyle = 'dashed')

ax1.text(150, 330, '1', fontsize = 30)
ax1.text(250, 330, '2', fontsize = 30)
ax1.text(350, 330, '3', fontsize = 30)
ax1.text(550, 330, '4', fontsize = 30)
ax1.text(710, 330, '5', fontsize = 30)
ax1.text(820, 330, '6', fontsize = 30)

# plt.title('CF_08_20cst_AT')
ax1.grid()
ax1.set_xlim(0,900)
ax1.set_ylim(0,600)
ax1.legend(['Force'], fontsize = 20, bbox_to_anchor=(0.725, 0.95))
ax2.legend(['Displacement'], fontsize = 20, bbox_to_anchor=(0.81, 0.82))

ax1.set_xticks([])
ax1.set_yticks([])
ax2.set_yticks([])
plt.show()

```



In [7]: `CF_08_20cst_AT_001.iloc[:,0]`

Out[7]:

0	0.16
1	0.18
2	0.20
3	0.22
4	0.24
...	
46252	925.20
46253	925.22
46254	925.24
46255	925.26
46256	925.28

Name: Zeit_s, Length: 46257, dtype: float64

In [8]:

```
data = pd.DataFrame()
data = [CF_08_20cst_AT_001, CF_08_20cst_AT_002, CF_08_20cst_AT_003, CF_08_20cst_AT_004]
result = pd.concat(data, axis = 1)
data = result.dropna()
```

In [9]:

```
plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True
```

```
fig, ax1 = plt.subplots()

x = CF_08_100cst_AT_001.iloc[:,0]
y1 = CF_08_100cst_AT_001.iloc[:,1]
y2 = CF_08_100cst_AT_001.iloc[:,2]
y2 = 6 - y2

x2 = CF_08_100cst_AT_002.iloc[:,0]
y3 = CF_08_100cst_AT_002.iloc[:,1]
y4 = CF_08_100cst_AT_002.iloc[:,2]
y4 = 6 - y4

x3 = CF_08_100cst_AT_003.iloc[:,0]
y5 = CF_08_100cst_AT_003.iloc[:,1]
y6 = CF_08_100cst_AT_003.iloc[:,2]
y6 = 6 - y6

x4 = CF_08_100cst_AT_004.iloc[:,0]
```

```

y7 = CF_08_100cst_AT_004.iloc[:,1]
y8 = CF_08_100cst_AT_004.iloc[:,2]
y8 = 6 - y8

color = 'red'

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x, y1, color=color, marker = 's', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x2, y3, color=color, marker = 'o', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x3, y5, color=color, marker = '^', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x4, y7, color=color, marker = '*', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'blue'
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x, y2, color=color, marker = 's', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x2, y4, color=color, marker = 'o', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

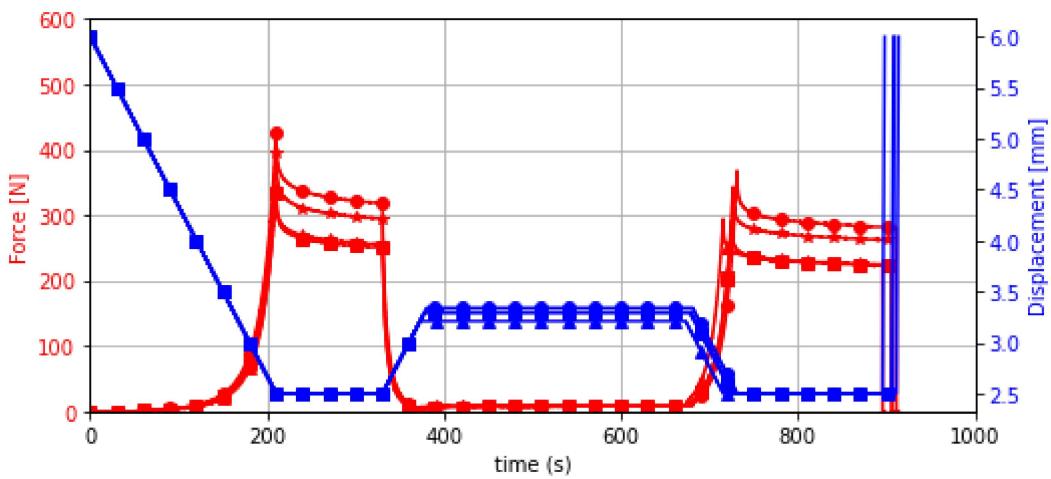
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x3, y6, color=color, marker = '^', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x4, y8, color=color, marker = '*', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

# plt.title('CF_08_100cst_AT')
ax1.set_xlim(0,600)
ax1.set_ylim(0,1000)
ax1.grid()
plt.show()

fig.savefig('CF_08_100cst_AT')

```



```
In [10]: plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True

fig, ax1 = plt.subplots()

x = CF_08_Dry_AT_001.iloc[:,0]
y1 = CF_08_Dry_AT_001.iloc[:,1]
y2 = CF_08_Dry_AT_001.iloc[:,2]
y2 = 6 - y2

x2 = CF_08_Dry_AT_002.iloc[:,0]
y3 = CF_08_Dry_AT_002.iloc[:,1]
y4 = CF_08_Dry_AT_002.iloc[:,2]
y4 = 6 - y4

x3 = CF_08_Dry_AT_003.iloc[:,0]
y5 = CF_08_Dry_AT_003.iloc[:,1]
y6 = CF_08_Dry_AT_003.iloc[:,2]
y6 = 6 - y6

x4 = CF_08_Dry_AT_004.iloc[:,0]
y7 = CF_08_Dry_AT_004.iloc[:,1]
y8 = CF_08_Dry_AT_004.iloc[:,2]
y8 = 6 - y8

color = 'red'

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x, y1, color=color, marker = 's', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x2, y3, color=color, marker = 'o', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

#ax1.set_xlabel('time (s)')
#ax1.set_ylabel('Force [N]', color=color)
#ax1.plot(x3, y5, color=color, marker = '^', markevery = 1500)
#ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
```

```

ax1.plot(x4, y7, color=color, marker = '*', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'blue'
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x, y2, color=color, marker = 's', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x2, y4, color=color, marker = 'o', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

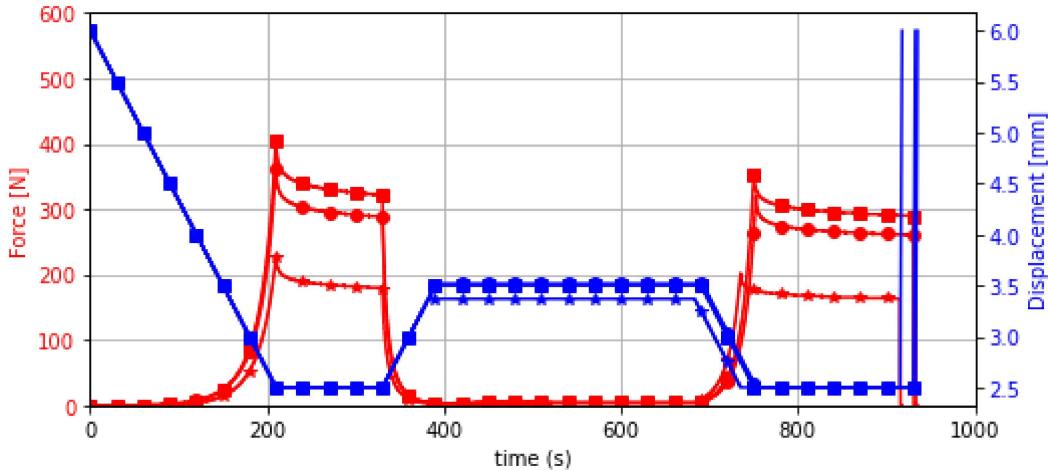
#ax2.set_ylabel('Displacement [mm]', color=color)
#ax2.plot(x3, y6, color=color, marker = '^', markevery = 1500)
#ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x4, y8, color=color, marker = '*', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

plt.title('CF_08_Dry_AT')
ax1.set_xlim(0,600)
ax1.set_ylim(0,600)
ax1.grid()
plt.show()

fig.savefig('CF_08_Dry_AT')

```



```

In [11]: plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True

fig, ax1 = plt.subplots()

x = CF_08_Dry_IT_001.iloc[:,0]
y1 = CF_08_Dry_IT_001.iloc[:,1]
y2 = CF_08_Dry_IT_001.iloc[:,2]
y2 = 6 - y2

x2 = CF_08_Dry_IT_002.iloc[:,0]
y3 = CF_08_Dry_IT_002.iloc[:,1]
y4 = CF_08_Dry_IT_002.iloc[:,2]
y4 = 6 - y4

```

```

x3 = CF_08_Dry_IT_003.iloc[:,0]
y5 = CF_08_Dry_IT_003.iloc[:,1]
y6 = CF_08_Dry_IT_003.iloc[:,2]
y6 = 6 - y6

x4 = CF_08_Dry_IT_004.iloc[:,0]
y7 = CF_08_Dry_IT_004.iloc[:,1]
y8 = CF_08_Dry_IT_004.iloc[:,2]
y8 = 6 - y8

color = 'red'

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x, y1, color=color, marker = 's', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x2, y3, color=color, marker = 'o', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x3, y5, color=color, marker = '^', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x4, y7, color=color, marker = '*', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'blue'
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x, y2, color=color, marker = 's', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x2, y4, color=color, marker = 'o', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

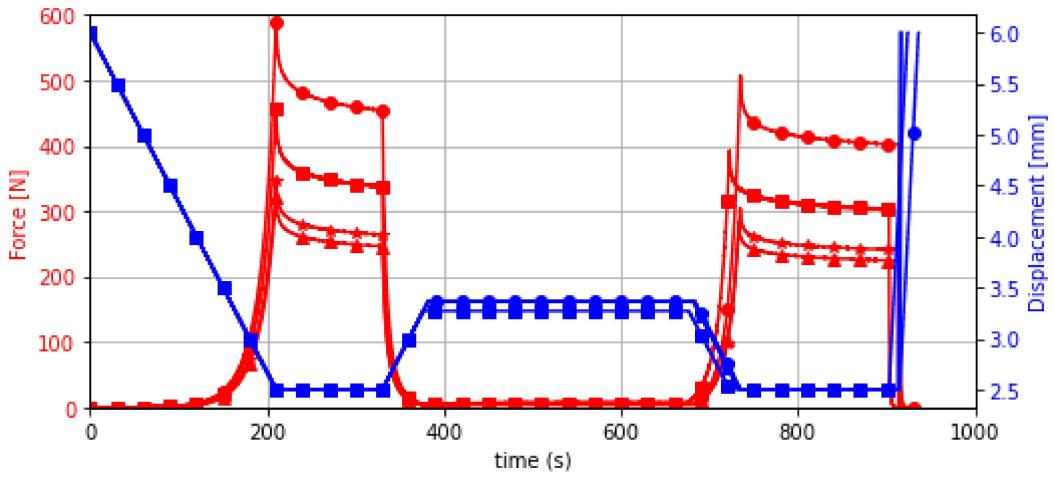
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x3, y6, color=color, marker = '^', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x4, y8, color=color, marker = '*', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

# plt.title('CF_08_Dry_IT')
ax1.set_xlim(0,1000)
ax1.set_ylim(0,600)
ax1.grid()
plt.show()

fig.savefig('CF_08_Dry_IT')

```



```
In [12]: plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True

fig, ax1 = plt.subplots()

x = CF_0_45_4_20cst_AT_001.iloc[:,0]
y1 = CF_0_45_4_20cst_AT_001.iloc[:,1]
y2 = CF_0_45_4_20cst_AT_001.iloc[:,2]
y2 = 6 - y2

x2 = CF_0_45_4_20cst_AT_002.iloc[:,0]
y3 = CF_0_45_4_20cst_AT_002.iloc[:,1]
y4 = CF_0_45_4_20cst_AT_002.iloc[:,2]
y4 = 6 - y4

x3 = CF_0_45_4_20cst_AT_003.iloc[:,0]
y5 = CF_0_45_4_20cst_AT_003.iloc[:,1]
y6 = CF_0_45_4_20cst_AT_003.iloc[:,2]
y6 = 6 - y6

x4 = CF_0_45_4_20cst_AT_004.iloc[:,0]
y7 = CF_0_45_4_20cst_AT_004.iloc[:,1]
y8 = CF_0_45_4_20cst_AT_004.iloc[:,2]
y8 = 6 - y8

color = 'red'

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x, y1, color=color, marker = 's', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x2, y3, color=color, marker = 'o', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x3, y5, color=color, marker = '^', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
```

```

ax1.plot(x4, y7, color=color, marker = '*', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'blue'
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x, y2, color=color, marker = 's', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x2, y4, color=color, marker = 'o', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

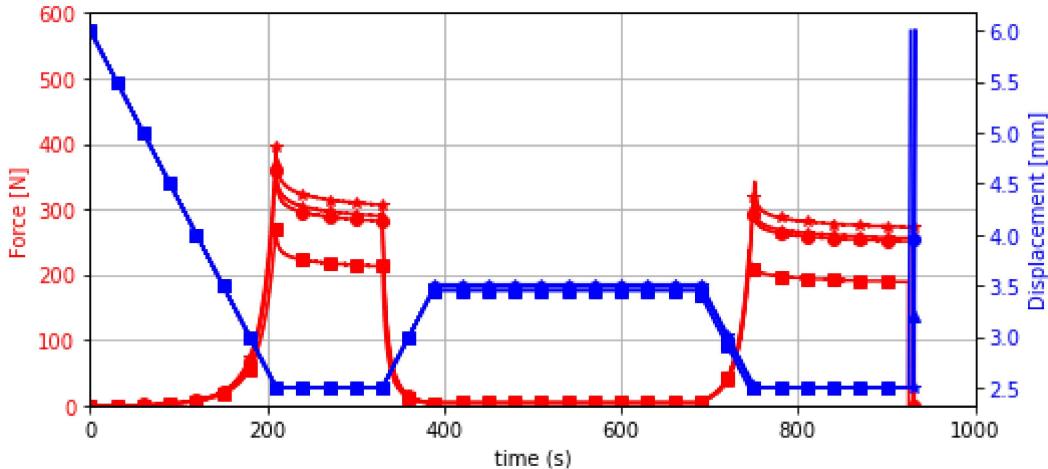
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x3, y6, color=color, marker = '^', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x4, y8, color=color, marker = '*', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

#plt.title('CF_0_45_4_20cst_AT')
ax1.set_xlim(0,1000)
ax1.set_ylim(0,600)
ax1.grid()
plt.show()

fig.savefig('CF_0_45_4_20cst_AT')

```



```

In [13]: plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True

fig, ax1 = plt.subplots()

x  = CF_0_45_4_100cst_AT_001.iloc[:,0]
y1 = CF_0_45_4_100cst_AT_001.iloc[:,1]
y2 = CF_0_45_4_100cst_AT_001.iloc[:,2]
y2 = 6 - y2

x2 = CF_0_45_4_100cst_AT_002.iloc[:,0]
y3 = CF_0_45_4_100cst_AT_002.iloc[:,1]
y4 = CF_0_45_4_100cst_AT_002.iloc[:,2]
y4 = 6 - y4

```

```

x3 = CF_0_45_4_100cst_AT_003.iloc[:,0]
y5 = CF_0_45_4_100cst_AT_003.iloc[:,1]
y6 = CF_0_45_4_100cst_AT_003.iloc[:,2]
y6 = 6 - y6

x4 = CF_0_45_4_100cst_AT_004.iloc[:,0]
y7 = CF_0_45_4_100cst_AT_004.iloc[:,1]
y8 = CF_0_45_4_100cst_AT_004.iloc[:,2]
y8 = 6 - y8

color = 'red'

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x, y1, color=color, marker = 's', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x2, y3, color=color, marker = 'o', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x3, y5, color=color, marker = '^', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x4, y7, color=color, marker = '*', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'blue'
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x, y2, color=color, marker = 's', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x2, y4, color=color, marker = 'o', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

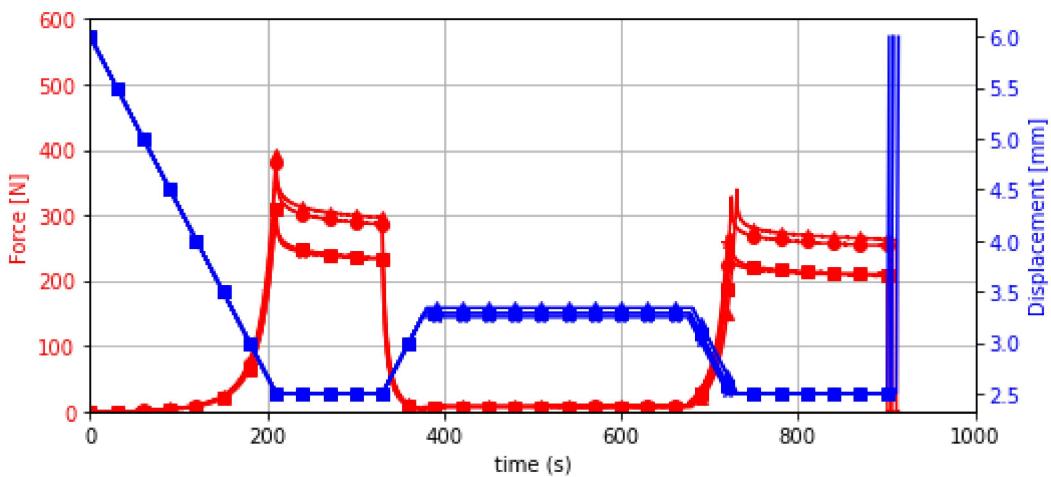
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x3, y6, color=color, marker = '^', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x4, y8, color=color, marker = '*', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

# plt.title('CF_0_45_4_100cst_AT')
ax1.set_xlim(0,1000)
ax1.set_ylim(0,600)
ax1.grid()
plt.show()

fig.savefig('CF_0_45_4_100cst_AT')

```



```
In [14]: plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True

fig, ax1 = plt.subplots()

x  = CF_0_45_4_Dry_AT_001.iloc[:,0]
y1 = CF_0_45_4_Dry_AT_001.iloc[:,1]
y2 = CF_0_45_4_Dry_AT_001.iloc[:,2]
y2 = 6 - y2

x2 = CF_0_45_4_Dry_AT_002.iloc[:,0]
y3 = CF_0_45_4_Dry_AT_002.iloc[:,1]
y4 = CF_0_45_4_Dry_AT_002.iloc[:,2]
y4 = 6 - y4

x3 = CF_0_45_4_Dry_AT_003.iloc[:,0]
y5 = CF_0_45_4_Dry_AT_003.iloc[:,1]
y6 = CF_0_45_4_Dry_AT_003.iloc[:,2]
y6 = 6 - y6

x4 = CF_0_45_4_Dry_AT_004.iloc[:,0]
y7 = CF_0_45_4_Dry_AT_004.iloc[:,1]
y8 = CF_0_45_4_Dry_AT_004.iloc[:,2]
y8 = 6 - y8

color = 'red'

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x, y1, color=color, marker = 's', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x2, y3, color=color, marker = 'o', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x3, y5, color=color, marker = '^', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
```

```

ax1.plot(x4, y7, color=color, marker = '*', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'blue'
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x, y2, color=color, marker = 's', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x2, y4, color=color, marker = 'o', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

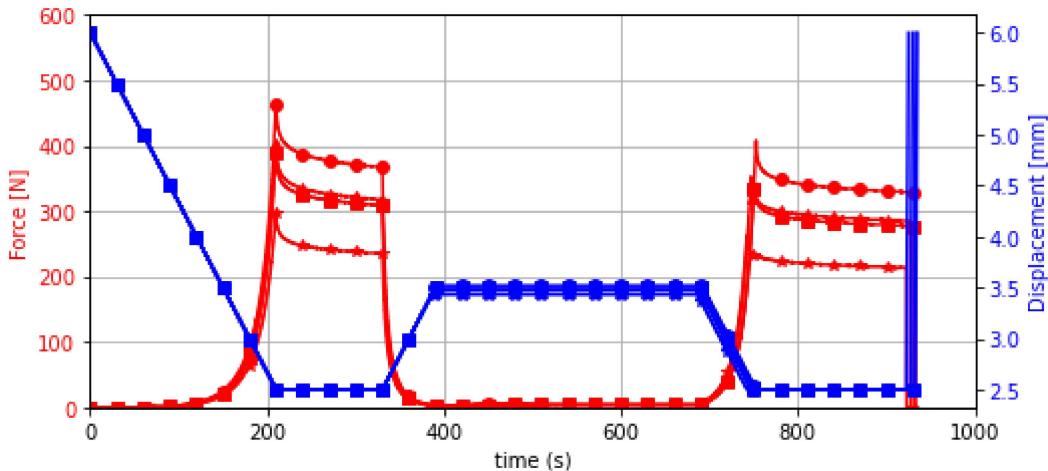
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x3, y6, color=color, marker = '^', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x4, y8, color=color, marker = '*', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

#plt.title('CF_0_45_4_Dry_AT')
ax1.set_xlim(0,1000)
ax1.set_ylim(0,600)
ax1.grid()
plt.show()

fig.savefig('CF_0_45_4_Dry_AT')

```



```

In [15]: plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True

fig, ax1 = plt.subplots()

x = CF_0_45_4_Dry_IT_001.iloc[:,0]
y1 = CF_0_45_4_Dry_IT_001.iloc[:,1]
y2 = CF_0_45_4_Dry_IT_001.iloc[:,2]
y2 = 6 - y2

x2 = CF_0_45_4_Dry_IT_002.iloc[:,0]
y3 = CF_0_45_4_Dry_IT_002.iloc[:,1]
y4 = CF_0_45_4_Dry_IT_002.iloc[:,2]
y4 = 6 - y4

```

```

x3 = CF_0_45_4_Dry_IT_003.iloc[:,0]
y5 = CF_0_45_4_Dry_IT_003.iloc[:,1]
y6 = CF_0_45_4_Dry_IT_003.iloc[:,2]
y6 = 6 - y6

x4 = CF_0_45_4_Dry_IT_004.iloc[:,0]
y7 = CF_0_45_4_Dry_IT_004.iloc[:,1]
y8 = CF_0_45_4_Dry_IT_004.iloc[:,2]
y8 = 6 - y8

color = 'red'

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x, y1, color=color, marker = 's', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x2, y3, color=color, marker = 'o', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x3, y5, color=color, marker = '^', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax1.set_xlabel('time (s)')
ax1.set_ylabel('Force [N]', color=color)
ax1.plot(x4, y7, color=color, marker = '*', markevery = 1500)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'blue'
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x, y2, color=color, marker = 's', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x2, y4, color=color, marker = 'o', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

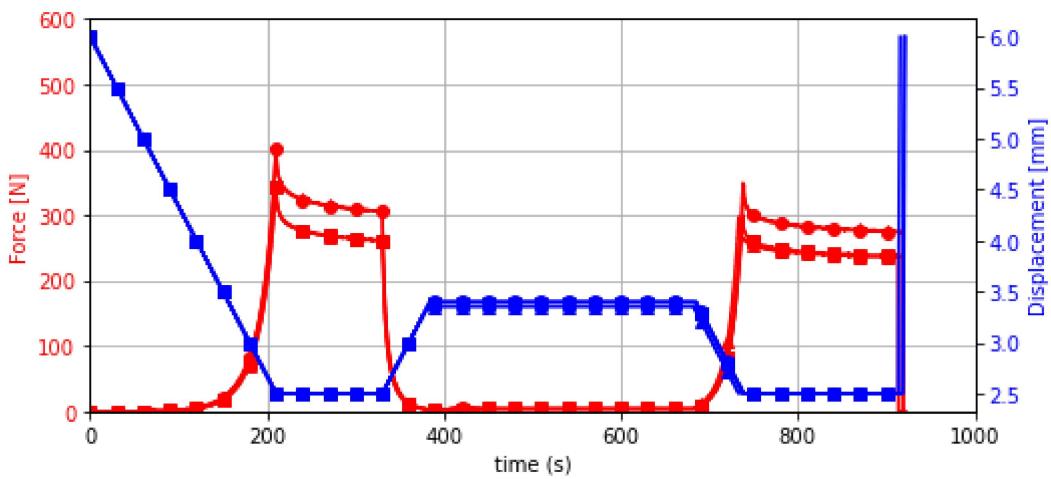
ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x3, y6, color=color, marker = '^', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

ax2.set_ylabel('Displacement [mm]', color=color)
ax2.plot(x4, y8, color=color, marker = '*', markevery = 1500)
ax2.tick_params(axis='y', labelcolor=color)

# plt.title('CF_0_45_4_Dry_IT')
ax1.set_xlim(0,1000)
ax1.set_ylim(0,600)
ax1.grid()
plt.show()

fig.savefig('CF_0_45_4_Dry_IT')

```



```
In [16]: def compaction(specimen1, specimen2, specimen3, specimen4):
    vector = np.zeros(int(data.shape[1]/3), dtype = int)
    n = 1
    for i in range(int(data.shape[1]/3)):
        vector[i] = n + i*3

    vector = vector.tolist()
    displacement = data.iloc[2:, vector]
    displacement_average = np.mean(displacement, axis = 1)

    vector = np.zeros(int(data.shape[1]/3), dtype = int)
    n = 2
    for i in range(int(data.shape[1]/3)):
        vector[i] = n + i*3

    force = data.iloc[2:, vector]
    force_average = np.mean(force, axis = 1) - friction
    force_average[force_average < 0] = 0
    standard_deviation = np.std(force, axis = 1)
    std_range = (force_average - standard_deviation, force_average + standard_deviation)

    return displacement_average, force_average, std_range
```

```
In [17]: CF_0_45_4_20cst_AT_004
```

Out[17]:

	Zeit_s	Kraft_N	Weg_mm
0	0.16	0	0.0
1	0.18	0	0.0
2	0.20	0	0.0
3	0.22	0	0.0
4	0.24	0	0.0
...
46661	933.38	0	0.0
46662	933.40	0	0.0
46663	933.42	0	0.0
46664	933.44	0	0.0
46665	933.46	0	0.0

46666 rows × 3 columns

```
In [18]: numRows = min(CF_0_45_4_20cst_AT_001.iloc[:, 0].size, CF_0_45_4_20cst_AT_002.iloc[:, 0].size)
numCols = 4

# Time dataframe
time = pd.DataFrame(index=range(numRows), columns=range(numCols))
time.iloc[:, 0] = CF_0_45_4_20cst_AT_001.iloc[0:numRows, 0]
time.iloc[:, 1] = CF_0_45_4_20cst_AT_002.iloc[0:numRows, 0]
time.iloc[:, 2] = CF_0_45_4_20cst_AT_003.iloc[0:numRows, 0]
time.iloc[:, 3] = CF_0_45_4_20cst_AT_004.iloc[0:numRows, 0]

## mean value
time_average = np.mean(time, axis = 1)

# Force dataframe
force = pd.DataFrame(index=range(numRows), columns=range(numCols))
force.iloc[:, 0] = CF_0_45_4_20cst_AT_001.iloc[0:numRows, 1]
force.iloc[:, 1] = CF_0_45_4_20cst_AT_002.iloc[0:numRows, 1]
force.iloc[:, 2] = CF_0_45_4_20cst_AT_003.iloc[0:numRows, 1]
force.iloc[:, 3] = CF_0_45_4_20cst_AT_004.iloc[0:numRows, 1]

## mean value and std
force_mean = np.mean(force, axis = 1)
force_std = np.std(force, axis = 1)
force_std_range = (force_mean - force_std, force_mean + force_std)

# Displacement dataframe
displacement = pd.DataFrame(index=range(numRows), columns=range(numCols))
displacement.iloc[:, 0] = CF_0_45_4_20cst_AT_001.iloc[0:numRows, 2]
displacement.iloc[:, 1] = CF_0_45_4_20cst_AT_002.iloc[0:numRows, 2]
displacement.iloc[:, 2] = CF_0_45_4_20cst_AT_003.iloc[0:numRows, 2]
displacement.iloc[:, 3] = CF_0_45_4_20cst_AT_004.iloc[0:numRows, 2]

## mean value and std
displacement_mean = 6 - np.mean(displacement, axis = 1)
displacement_std = np.std(displacement, axis = 1)
```

```
displacement_std_range = (displacement_mean - displacement_std, displacement_mean + di  
s = {'Time': time_average, 'Force-mean': force_mean, 'Force-std': force_std_range, 'Di
```

```
In [19]: def compaction(specimen1, specimen2, specimen3, specimen4):  
  
    numRows = min(specimen1.iloc[:, 0].size, specimen2.iloc[:, 0].size, specimen3.iloc[:, 0].size, specimen4.iloc[:, 0].size)  
    numCols = 4  
  
    # Time dataframe  
    time = pd.DataFrame(index=range(numRows), columns=range(numCols))  
    time.iloc[:, 0] = specimen1.iloc[0:numRows, 0]  
    time.iloc[:, 1] = specimen2.iloc[0:numRows, 0]  
    time.iloc[:, 2] = specimen3.iloc[0:numRows, 0]  
    time.iloc[:, 3] = specimen4.iloc[0:numRows, 0]  
  
    ## mean value  
    time_average = np.mean(time, axis = 1)  
  
    # Force dataframe  
    force = pd.DataFrame(index=range(numRows), columns=range(numCols))  
    force.iloc[:, 0] = specimen1.iloc[0:numRows, 1]  
    force.iloc[:, 1] = specimen2.iloc[0:numRows, 1]  
    force.iloc[:, 2] = specimen3.iloc[0:numRows, 1]  
    force.iloc[:, 3] = specimen4.iloc[0:numRows, 1]  
  
    ## mean value and std  
    force_mean = np.mean(force, axis = 1)  
    force_std = np.std(force, axis = 1)  
    force_std_range = (force_mean - force_std, force_mean + force_std)  
  
    # Displacement dataframe  
    displacement = pd.DataFrame(index=range(numRows), columns=range(numCols))  
    displacement.iloc[:, 0] = specimen1.iloc[0:numRows, 2]  
    displacement.iloc[:, 1] = specimen2.iloc[0:numRows, 2]  
    displacement.iloc[:, 2] = specimen3.iloc[0:numRows, 2]  
    displacement.iloc[:, 3] = specimen4.iloc[0:numRows, 2]  
  
    ## mean value and std  
    displacement_mean = 6 - np.mean(displacement, axis = 1)  
    displacement_std = np.std(displacement, axis = 1)  
    displacement_std_range = (displacement_mean - displacement_std, displacement_mean + displacement_std)  
  
    dictionary = {'Time': time_average, 'Force-mean': force_mean, 'Force-std': force_std_range, 'Displacement': displacement_mean, 'Displacement-std': displacement_std_range}  
  
    return dictionary
```

```
In [20]: CF_08_Dry_AT = compaction(CF_08_Dry_AT_001, CF_08_Dry_AT_002, CF_08_Dry_AT_003, CF_08_Dry_AT_004)  
CF_08_Dry_IT = compaction(CF_08_Dry_IT_001, CF_08_Dry_IT_002, CF_08_Dry_IT_003, CF_08_Dry_IT_004)  
CF_08_100cst_AT = compaction(CF_08_100cst_AT_001, CF_08_100cst_AT_002, CF_08_100cst_AT_003, CF_08_100cst_AT_004)  
CF_08_20cst_AT = compaction(CF_08_20cst_AT_001, CF_08_20cst_AT_002, CF_08_20cst_AT_003, CF_08_20cst_AT_004)  
  
CF_0_45_4_Dry_AT = compaction(CF_0_45_4_Dry_AT_001, CF_0_45_4_Dry_AT_002, CF_0_45_4_Dry_AT_003, CF_0_45_4_Dry_AT_004)  
CF_0_45_4_Dry_IT = compaction(CF_0_45_4_Dry_IT_001, CF_0_45_4_Dry_IT_002, CF_0_45_4_Dry_IT_003, CF_0_45_4_Dry_IT_004)  
CF_0_45_4_20cst_AT = compaction(CF_0_45_4_20cst_AT_001, CF_0_45_4_20cst_AT_002, CF_0_45_4_20cst_AT_003, CF_0_45_4_20cst_AT_004)  
CF_0_45_4_100cst_AT = compaction(CF_0_45_4_100cst_AT_001, CF_0_45_4_100cst_AT_002, CF_0_45_4_100cst_AT_003, CF_0_45_4_100cst_AT_004)
```

```
In [21]: def compaction_plot(dictionary1, label1, dictionary2, label2, name_figure):  
    fig, ax1 = plt.subplots(figsize = (24,12))
```

```

ax1.plot(dictionary1['Time'], dictionary1['Force-mean'], marker = 'o', markevery =
ax1.fill_between(dictionary1['Time'], dictionary1['Force-std'][0],dictionary1['For

ax1.plot(dictionary2['Time'], dictionary2['Force-mean'], marker = 's', markevery =
ax1.fill_between(dictionary2['Time'], dictionary2['Force-std'][0],dictionary2['For
plt.xticks(fontsize=30)
plt.yticks(fontsize=30)
ax2 = ax1.twinx()

ax2.plot(dictionary1['Time'], dictionary1['Displacement'], color='orange', marker =
ax2.fill_between(dictionary1['Time'], dictionary1['Displacement-std'][0],dictionar

ax2.plot(dictionary2['Time'], dictionary2['Displacement'], color='green', marker =
ax2.fill_between(dictionary2['Time'], dictionary2['Displacement-std'][0],dictionar

first_nonzero = next((i for i, x in enumerate(force_mean) if x), None)
ax1.legend(fontsize="30", loc='upper center', bbox_to_anchor=(0.5, -0.05),
          fancybox=True, shadow=True, ncol=2)

ax1.set_ylabel('Force [N]', fontsize=30)

ax2.legend(fontsize="30", loc='upper center', bbox_to_anchor=(0.5, -0.15),
          fancybox=True, shadow=True, ncol=2)
ax2.set_ylabel('Displacement [mm]', fontsize=30)
plt.yticks(fontsize=30)

#ax1.axvLine(x = time_average[first_nonzero], color = 'magenta', Label = 'axvLine'
ax1.set_xlim(0,1000)
ax1.set_ylim(0,600)
ax1.grid()
fig.savefig(name_figure + '.png')

```

In [22]:

```

def compaction_plot3(dictionary1, label1, dictionary2, label2, dictionary3, label3, na
fig, ax1 = plt.subplots(figsize = (30, 12))

ax1.plot(dictionary1['Time'], dictionary1['Force-mean'], marker = 'o', markevery =
ax1.fill_between(dictionary1['Time'], dictionary1['Force-std'][0],dictionary1['For

ax1.plot(dictionary2['Time'], dictionary2['Force-mean'], marker = 's', markevery =
ax1.fill_between(dictionary2['Time'], dictionary2['Force-std'][0],dictionary2['For

ax1.plot(dictionary3['Time'], dictionary3['Force-mean'], marker = '^', markevery =
ax1.fill_between(dictionary3['Time'], dictionary3['Force-std'][0],dictionary3['For

plt.xticks(fontsize=30)
plt.yticks(fontsize=30)
ax2 = ax1.twinx()

ax2.plot(dictionary1['Time'], dictionary1['Displacement'], color='orange', marker =
ax2.fill_between(dictionary1['Time'], dictionary1['Displacement-std'][0],dictionar

ax2.plot(dictionary2['Time'], dictionary2['Displacement'], color='green', marker =
ax2.fill_between(dictionary2['Time'], dictionary2['Displacement-std'][0],dictionar

ax2.plot(dictionary3['Time'], dictionary3['Displacement'], color='cadetblue', mark
ax2.fill_between(dictionary3['Time'], dictionary3['Displacement-std'][0],dictionar

```

```

first_nonzero = next((i for i, x in enumerate(force_mean) if x), None)
ax1.legend(fontsize="30", loc='upper center', bbox_to_anchor=(0.5, -0.05),
           fancybox=True, shadow=True, ncol=3)

ax1.set_ylabel('Force [N]', fontsize=30)

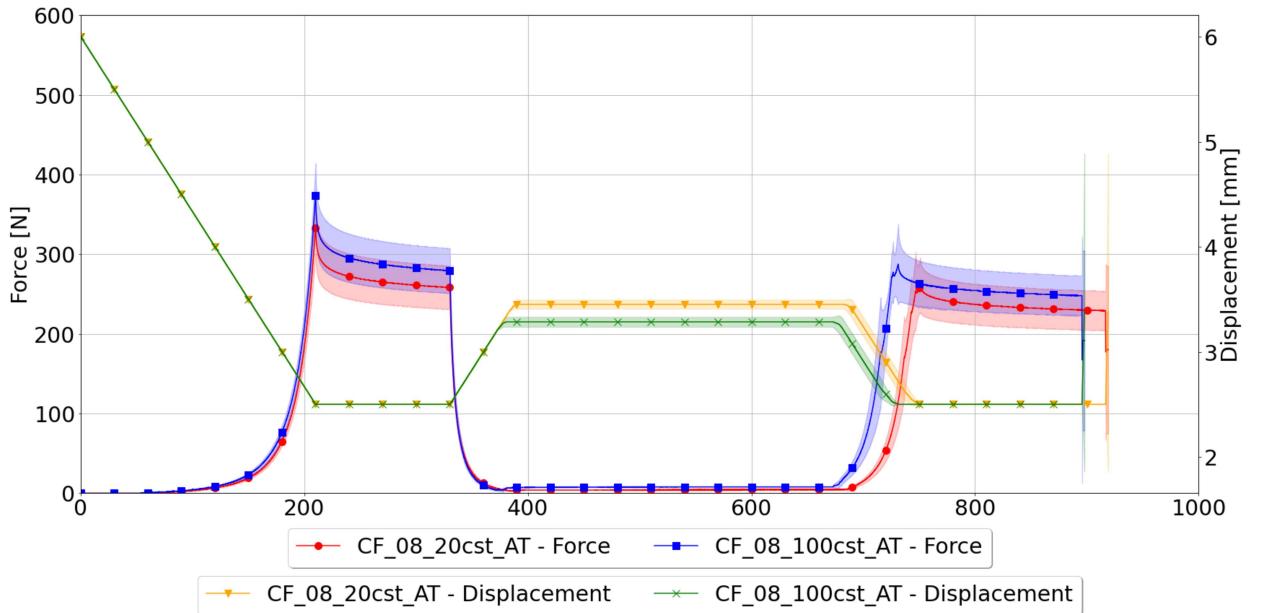
ax2.legend(fontsize="30", loc='upper center', bbox_to_anchor=(0.5, -0.15),
           fancybox=True, shadow=True, ncol=3)
ax2.set_ylabel('Displacement [mm]', fontsize=30)
plt.yticks(fontsize=30)

#ax1.axvline(x = time_average[first_nonzero], color = 'magenta', label = 'axvline')
ax1.set_xlim(0,1000)
ax1.set_ylim(0,600)
ax1.grid()

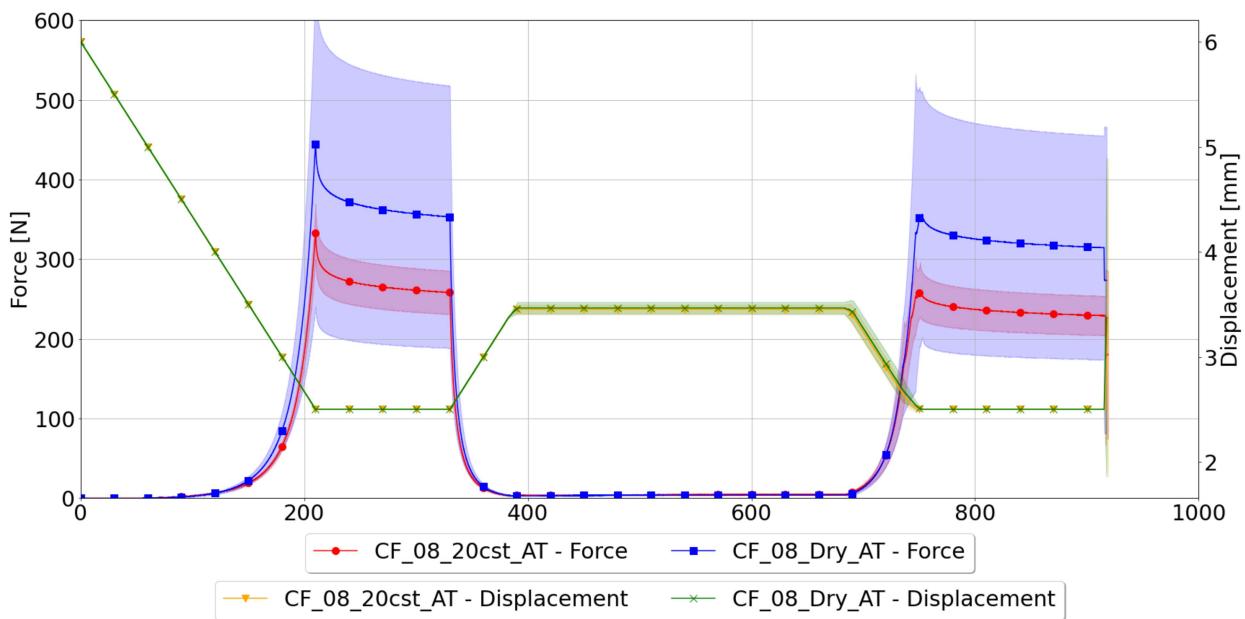
fig.savefig(name_figure + '.png')

```

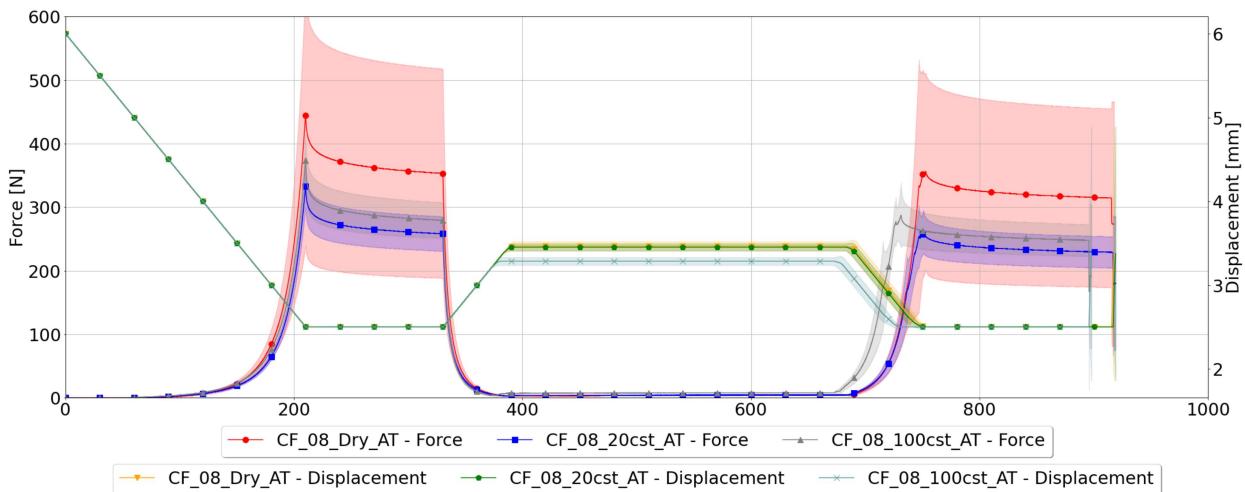
In [23]: `compaction_plot(CF_08_20cst_AT, 'CF_08_20cst_AT', CF_08_100cst_AT, 'CF_08_100cst_AT', '`



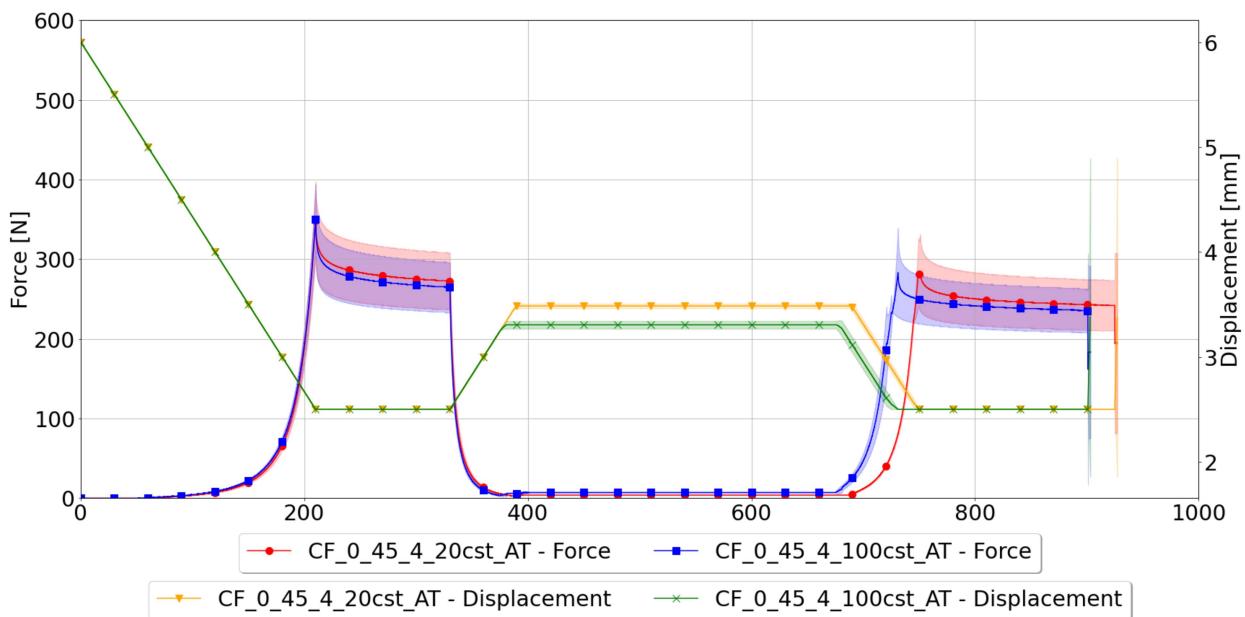
In [24]: `compaction_plot(CF_08_20cst_AT, 'CF_08_20cst_AT', CF_08_Dry_AT, 'CF_08_Dry_AT', 'apagar`



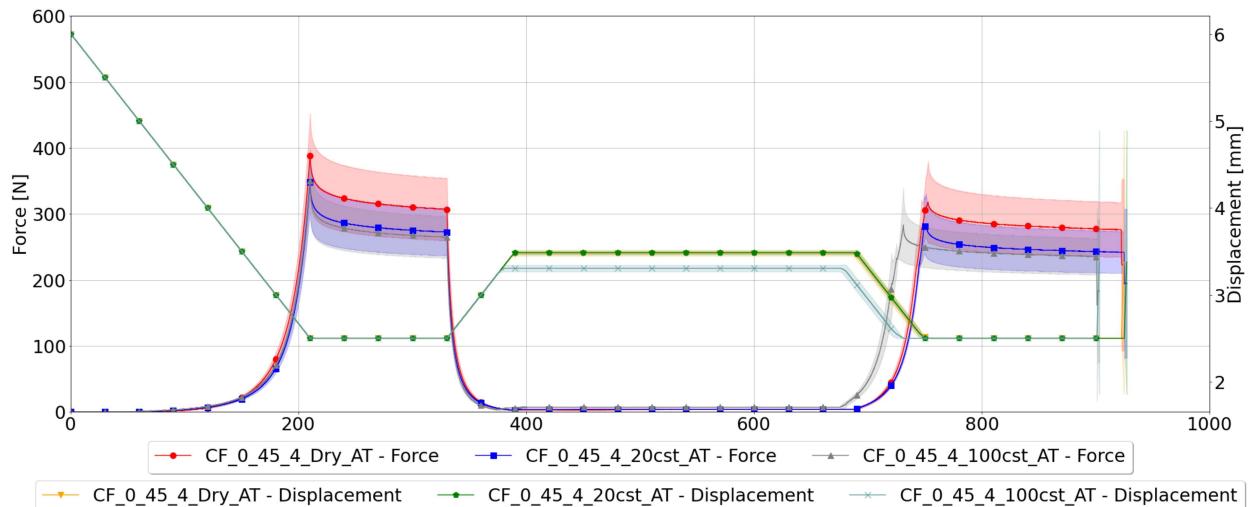
```
In [25]: compaction_plot3(CF_08_Dry_AT, 'CF_08_Dry_AT', CF_08_20cst_AT, 'CF_08_20cst_AT', CF_08_
```



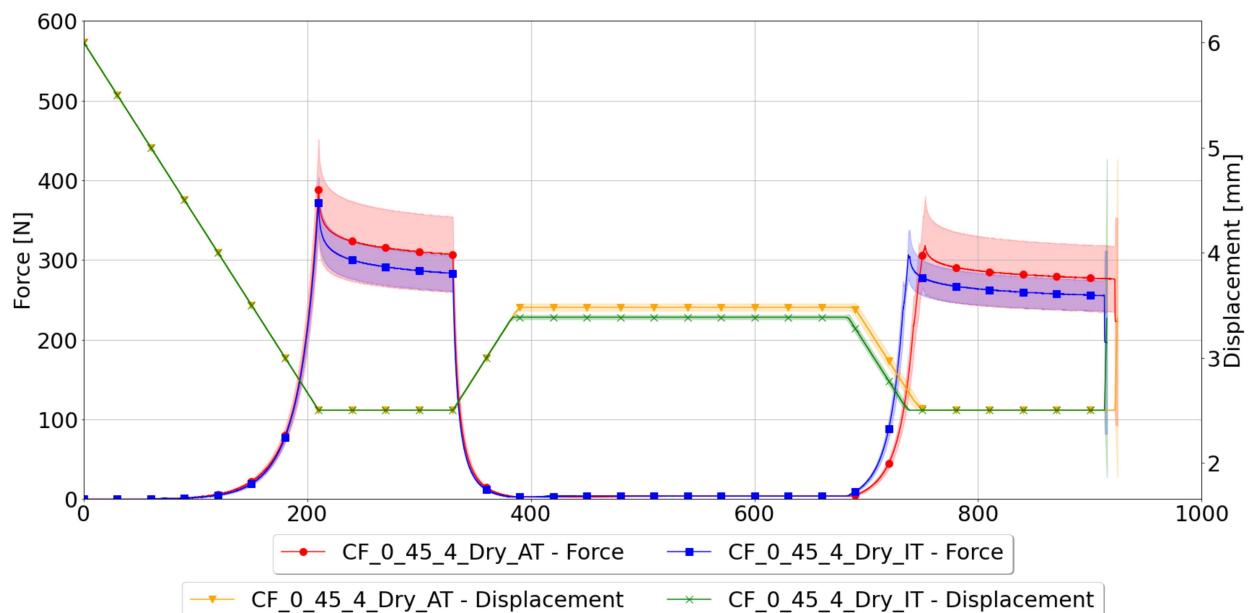
```
In [26]: compaction_plot(CF_0_45_4_20cst_AT, 'CF_0_45_4_20cst_AT', CF_0_45_4_100cst_AT, 'CF_0_45_4_100cst_AT')
```



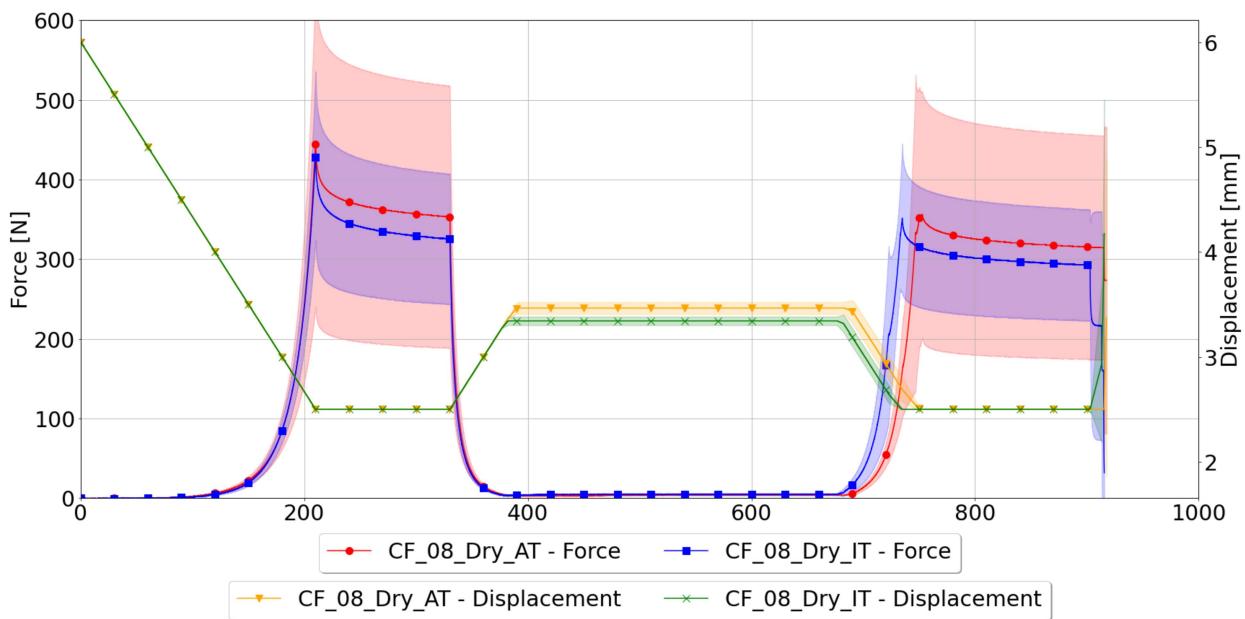
```
In [27]: compaction_plot3(CF_0_45_4_Dry_AT, 'CF_0_45_4_Dry_AT', CF_0_45_4_20cst_AT, 'CF_0_45_4_
```



```
In [28]: compaction_plot(CF_0_45_4_Dry_AT, 'CF_0_45_4_Dry_AT', CF_0_45_4_Dry_IT, 'CF_0_45_4_Dry
```



```
In [29]: compaction_plot(CF_08_Dry_AT, 'CF_08_Dry_AT', CF_08_Dry_IT, 'CF_08_Dry_IT', 'CF_08')
```



In [156...]

```
def plot_force_displacement(specimen_1, label1, specimen_2, label2, specimen_3, label3):

    # selecting rows specimen 1
    array = np.linspace(0,specimen_1['Force-mean'].size - 1,specimen_1['Force-mean'].size)
    specimen_1['index'] = array
    vector_1 = (specimen_1['Force-mean'] != 0) & (specimen_1['index'] < specimen_1['Force-mean'].size)
    initial_value = specimen_1['Displacement'][vector_1].reset_index()[0][0]
    specimen_1['Displacement'][vector_1] = np.absolute(initial_value - specimen_1['Displacement'][vector_1])

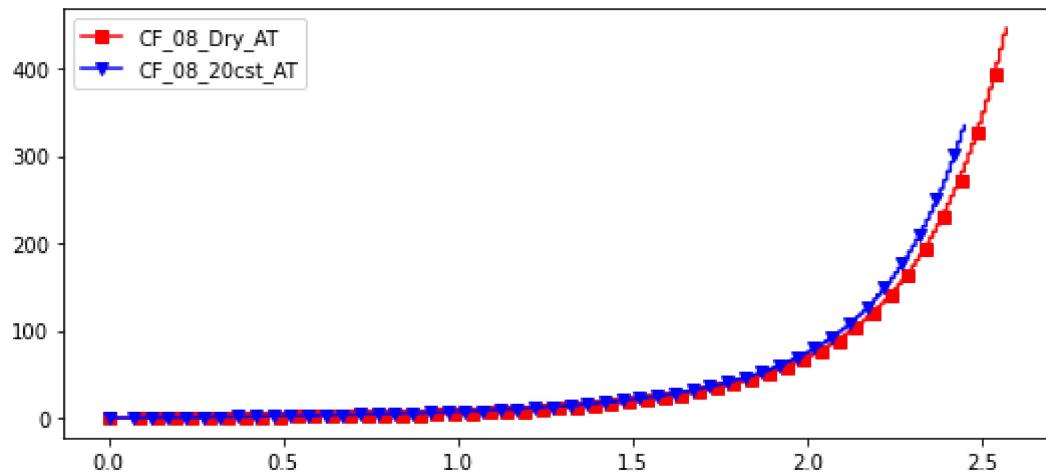
    # selecting rows specimen 2
    array = np.linspace(0,specimen_2['Force-mean'].size - 1,specimen_2['Force-mean'].size)
    specimen_2['index'] = array
    vector_2 = (specimen_2['Force-mean'] != 0) & (specimen_2['index'] < specimen_2['Force-mean'].size)
    initial_value = specimen_2['Displacement'][vector_2].reset_index()[0][0]
    specimen_2['Displacement'][vector_2] = np.absolute(initial_value - specimen_2['Displacement'][vector_2])

    # selecting rows specimen 3
    array = np.linspace(0,specimen_3['Force-mean'].size - 1,specimen_3['Force-mean'].size)
    specimen_3['index'] = array
    vector_3 = (specimen_3['Force-mean'] != 0) & (specimen_3['index'] < specimen_3['Force-mean'].size)
    initial_value = specimen_3['Displacement'][vector_3].reset_index()[0][0]
    specimen_3['Displacement'][vector_3] = np.absolute(initial_value - specimen_3['Displacement'][vector_3])

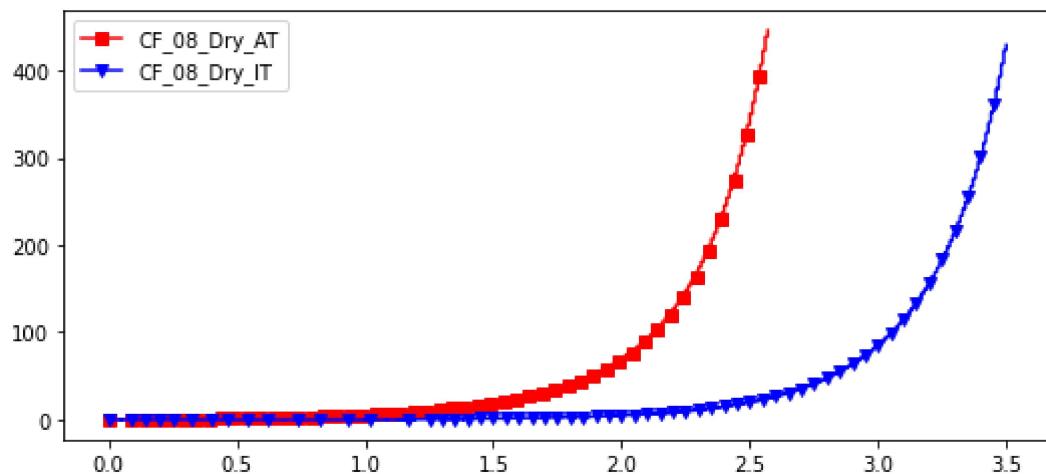
    # plot
    fig, ax1 = plt.subplots(figsize = (25, 10))
    ax1.set_ylabel('Force [N]', fontsize=30)
    ax1.set_xlabel('Displacement [mm]', fontsize=30)
    plt.xticks(fontsize=30)
    plt.yticks(fontsize=30)

    ax1.plot(specimen_1['Displacement'][vector_1], specimen_1['Force-mean'][vector_1], color='red')
    ax1.plot(specimen_2['Displacement'][vector_2], specimen_2['Force-mean'][vector_2], color='blue')
    ax1.plot(specimen_3['Displacement'][vector_3], specimen_3['Force-mean'][vector_3], color='orange')
    ax1.grid()
    ax1.set_xlim(0,3)
    ax1.set_ylim(0,500)
    ax1.legend(fontsize="30", loc='upper center', ncol=3)
```

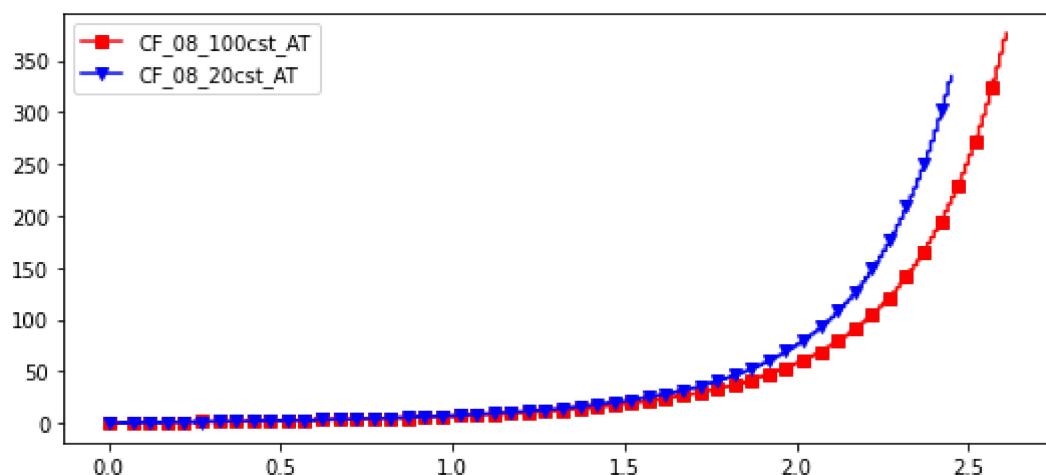
```
In [111]: plot_force_displacement(CF_08_Dry_AT, 'CF_08_Dry_AT', CF_08_20cst_AT, 'CF_08_20cst_AT')
```



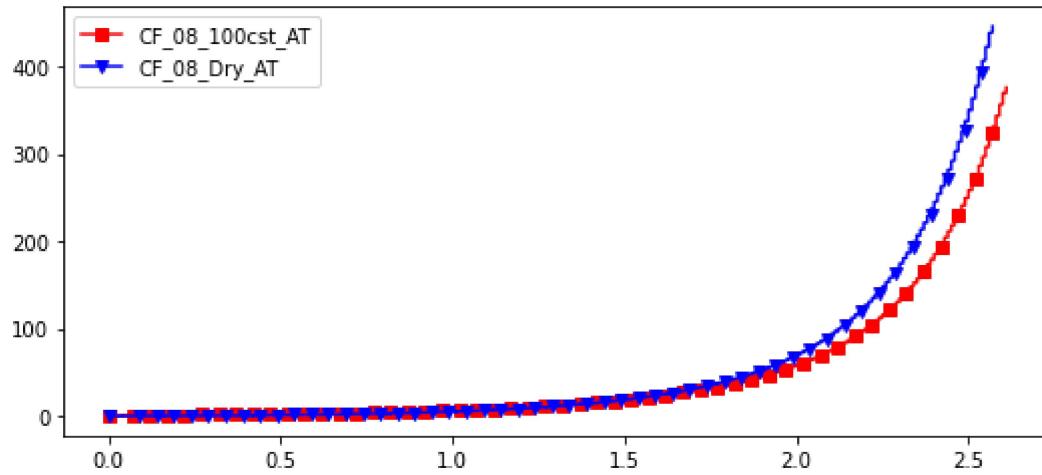
```
In [112]: plot_force_displacement(CF_08_Dry_AT, 'CF_08_Dry_AT', CF_08_Dry_IT, 'CF_08_Dry_IT')
```



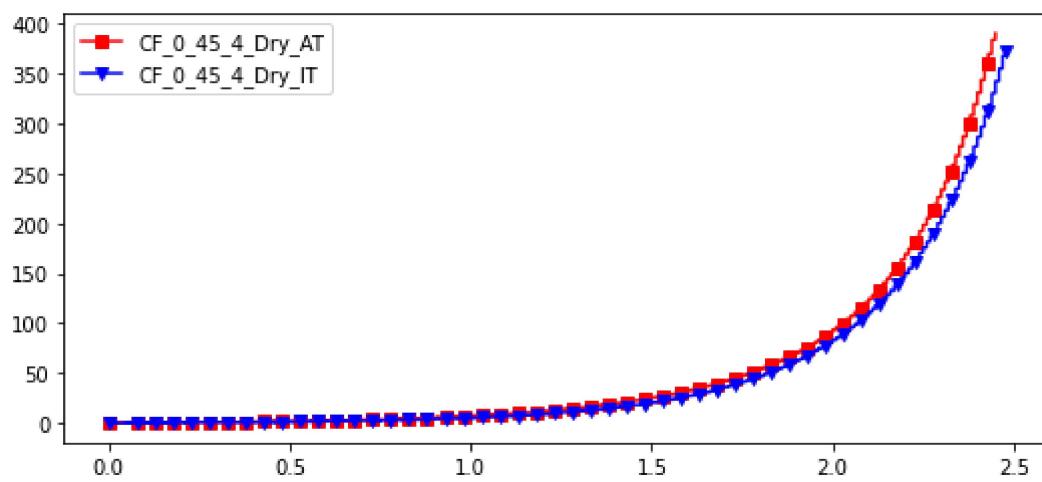
```
In [113]: plot_force_displacement(CF_08_100cst_AT, 'CF_08_100cst_AT', CF_08_20cst_AT, 'CF_08_20cst_AT')
```



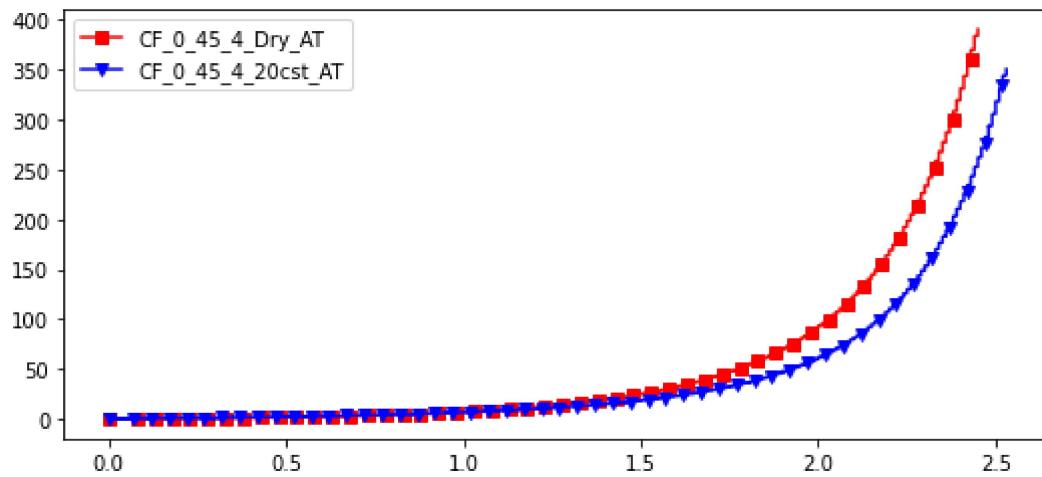
```
In [114]: plot_force_displacement(CF_08_100cst_AT, 'CF_08_100cst_AT', CF_08_Dry_AT, 'CF_08_Dry_AT')
```



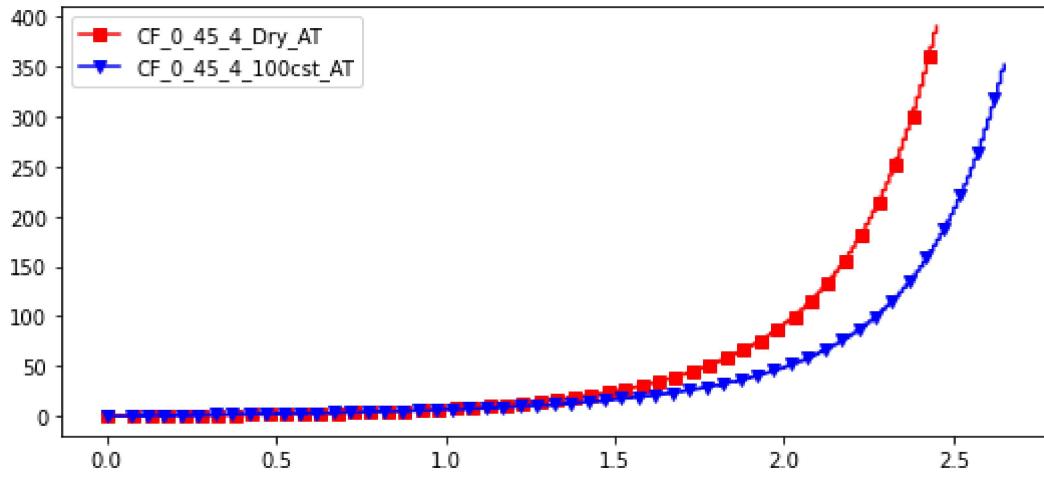
```
In [115]: plot_force_displacement(CF_0_45_4_Dry_AT, 'CF_0_45_4_Dry_AT', CF_0_45_4_Dry_IT, 'CF_0_45_4_Dry_IT')
```



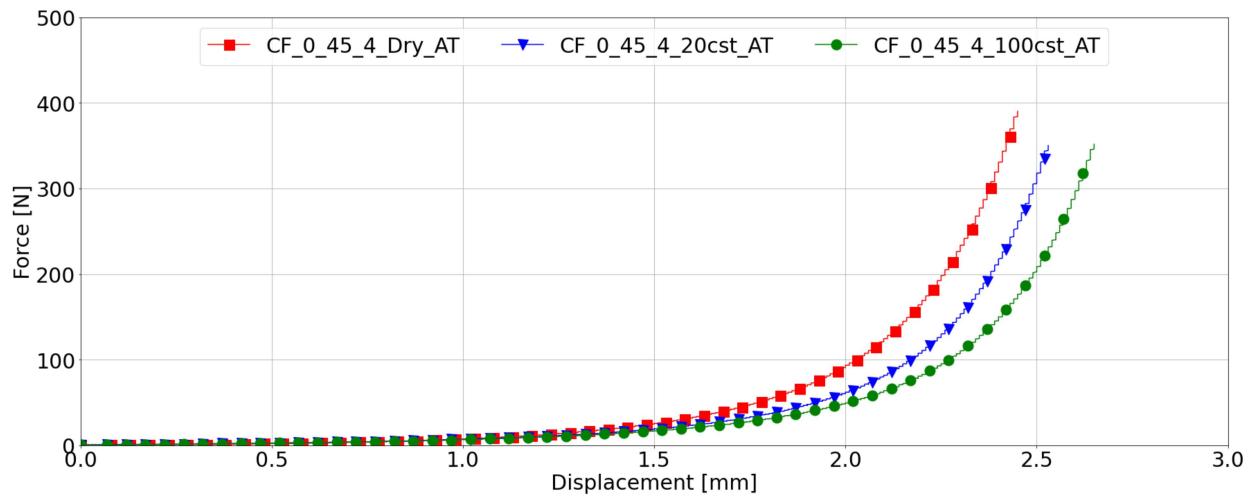
```
In [116]: plot_force_displacement(CF_0_45_4_Dry_AT, 'CF_0_45_4_Dry_AT', CF_0_45_4_20cst_AT, 'CF_0_45_4_20cst_AT')
```



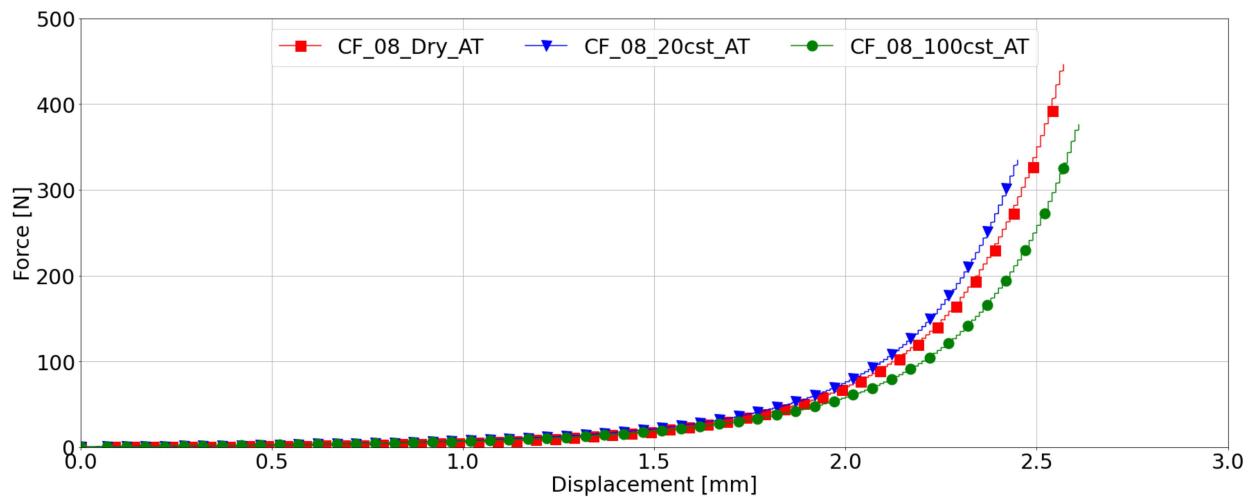
```
In [117]: plot_force_displacement(CF_0_45_4_Dry_AT, 'CF_0_45_4_Dry_AT', CF_0_45_4_100cst_AT, 'CF_0_45_4_100cst_AT')
```



```
In [157]: plot_force_displacement(CF_0_45_4_Dry_AT, 'CF_0_45_4_Dry_AT', CF_0_45_4_20cst_AT, 'CF_0_45_4_20cst_AT')
```



```
In [158]: plot_force_displacement(CF_08_Dry_AT, 'CF_08_Dry_AT', CF_08_20cst_AT, 'CF_08_20cst_AT', CF_08_100cst_AT, 'CF_08_100cst_AT')
```



```
In [133]: def plot_force_displacement_single(specimen_1, label1, specimen_2, label2, specimen_3)
```

```
    numRows = 10491

    vector_1 = np.zeros((1,specimen_1['Weg_mm'].size), dtype=int)
    vector_1[0,0:numRows] = 1
    vector_1 = np.ma.masked_equal(vector_1, 1)
```

```

vector_2 = np.zeros((1,specimen_2['Weg_mm'].size), dtype=int)
vector_2[0,0:numRows] = 1
vector_2 = np.ma.masked_equal(vector_2, 1)

if (not specimen_3.empty) & (specimen_4.empty):
    vector_3 = np.zeros((1,specimen_3['Weg_mm'].size), dtype=int)
    vector_3[0,0:numRows] = 1
    vector_3 = np.ma.masked_equal(vector_3, 1)

numCols = 3

elif (specimen_3.empty) & (not specimen_4.empty):
    vector_4 = np.zeros((1,specimen_4['Weg_mm'].size), dtype=int)
    vector_4[0,0:numRows] = 1
    vector_4 = np.ma.masked_equal(vector_4, 1)

    numCols = 3

elif (not specimen_3.empty) & (not specimen_4.empty):
    vector_3 = np.zeros((1,specimen_3['Weg_mm'].size), dtype=int)
    vector_3[0,0:numRows] = 1
    vector_3 = np.ma.masked_equal(vector_3, 1)

    vector_4 = np.zeros((1,specimen_4['Weg_mm'].size), dtype=int)
    vector_4[0,0:numRows] = 1
    vector_4 = np.ma.masked_equal(vector_4, 1)

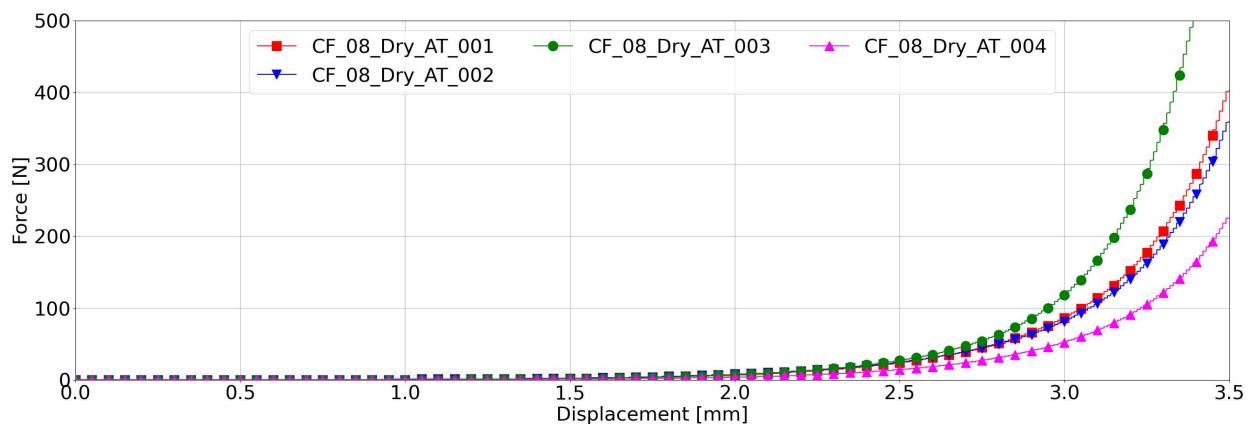
    numCols = 4
elif (specimen_3.empty) & (specimen_4.empty):
    numCols = 2

# plot
fig, ax1 = plt.subplots(figsize = (32, 10))
ax1.set_ylabel('Force [N]', fontsize=30)
ax1.set_xlabel('Displacement [mm]', fontsize=30)
plt.xticks(fontsize=30)
plt.yticks(fontsize=30)

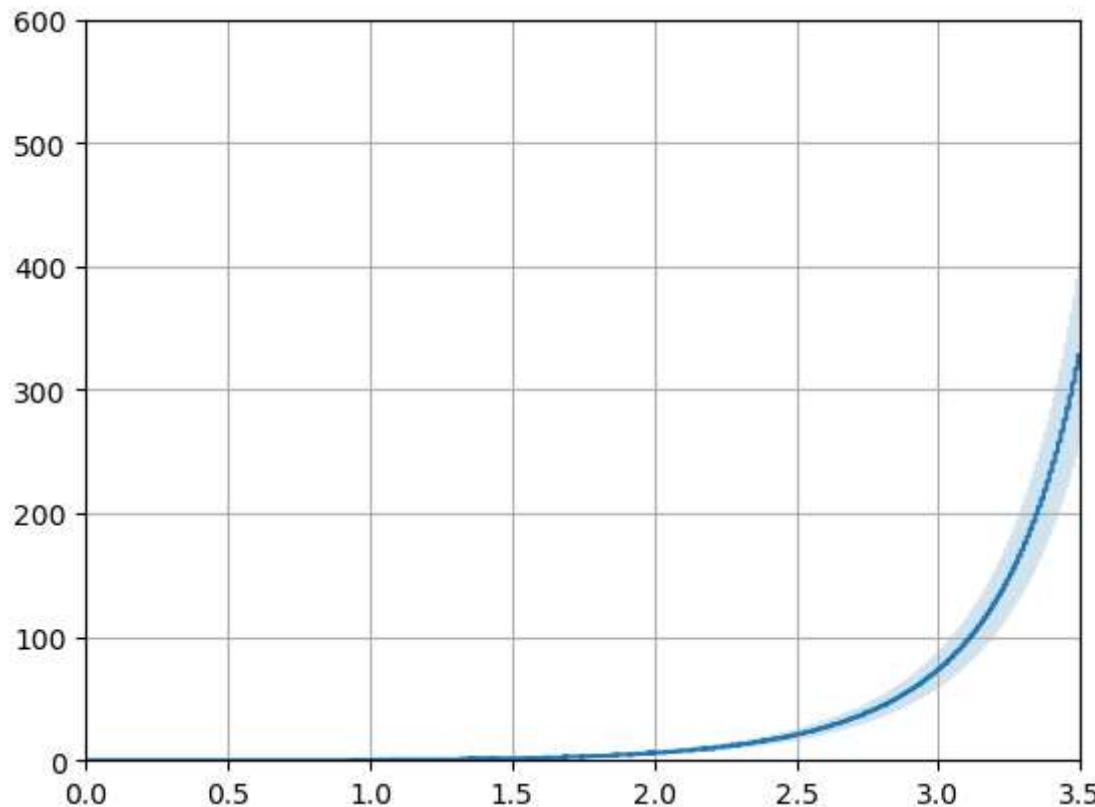
ax1.plot(specimen_1['Weg_mm'][vector_1.mask[0]], specimen_1['Kraft_N'][vector_1.mask[0]])
ax1.plot(specimen_2['Weg_mm'][vector_2.mask[0]], specimen_2['Kraft_N'][vector_2.mask[0]])
if (not specimen_3.empty):
    ax1.plot(specimen_3['Weg_mm'][vector_3.mask[0]], specimen_3['Kraft_N'][vector_3.mask[0]])
if (not specimen_4.empty):
    ax1.plot(specimen_4['Weg_mm'][vector_4.mask[0]], specimen_4['Kraft_N'][vector_4.mask[0]])
ax1.grid()
ax1.set_xlim(0,3.5)
ax1.set_ylim(0,500)
ax1.legend(fontsize="30", loc='upper center', ncol=3)

```

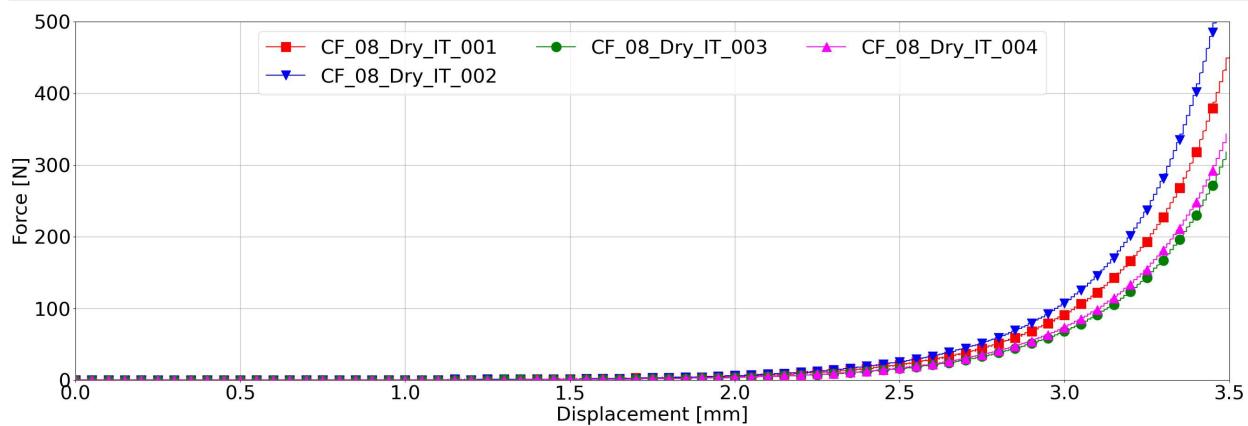
In [135]: plot_force_displacement_single(CF_08_Dry_AT_001, 'CF_08_Dry_AT_001', CF_08_Dry_AT_002,



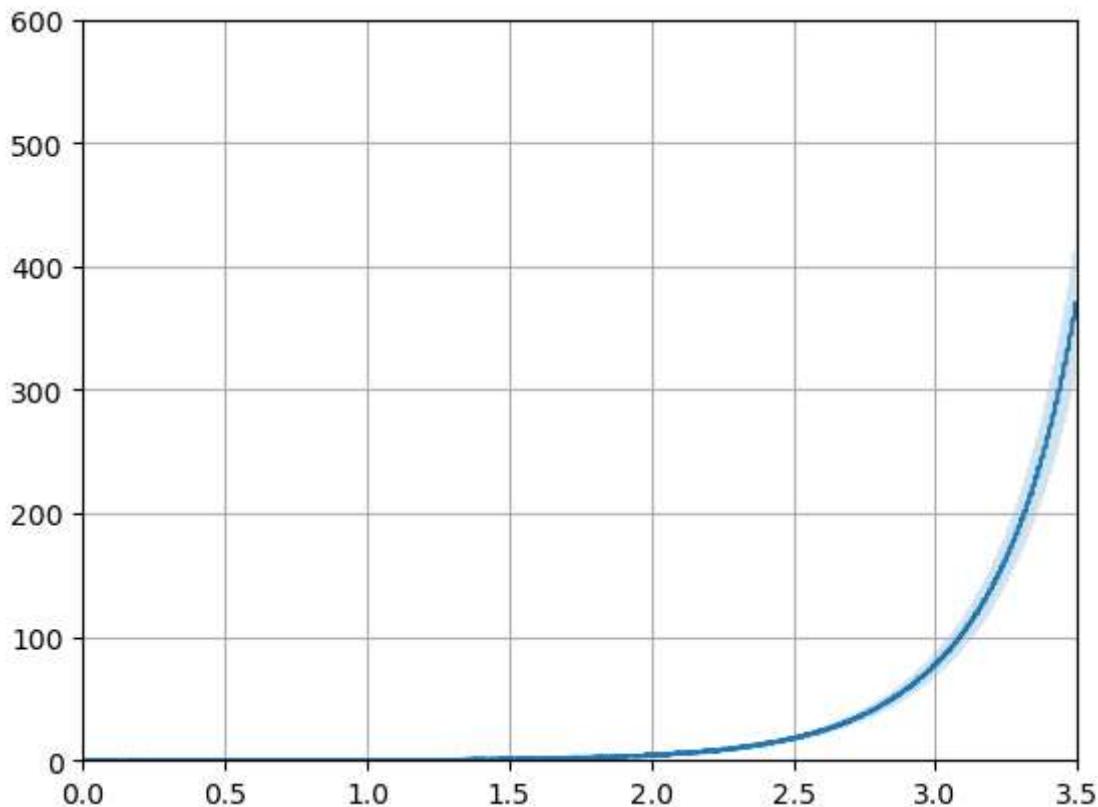
```
In [151...]: CF_08_Dry_AT = first_compaction(CF_08_Dry_AT_001, CF_08_Dry_AT_002, CF_08_Dry_AT_004)
```



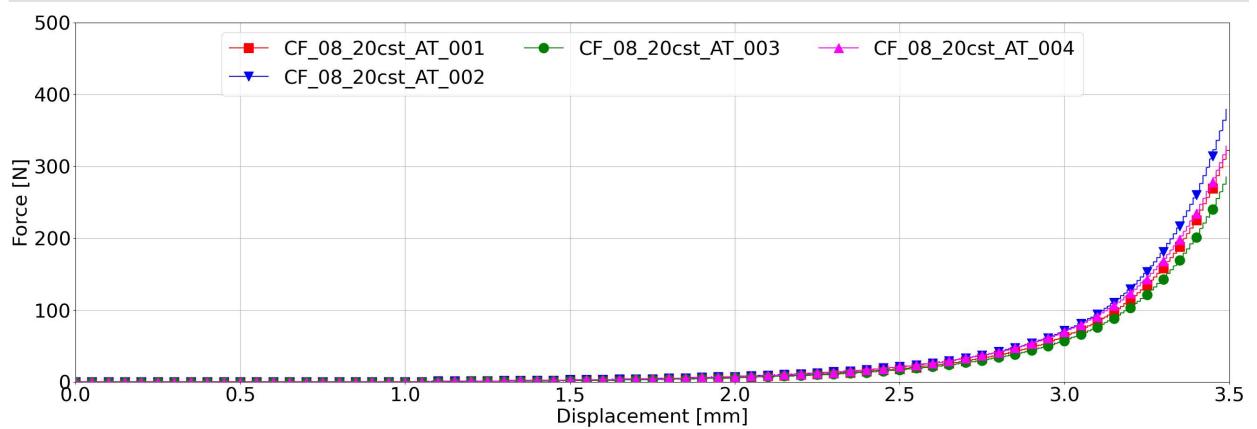
```
In [136...]: plot_force_displacement_single(CF_08_Dry_IT_001, 'CF_08_Dry_IT_001', CF_08_Dry_IT_002,
```



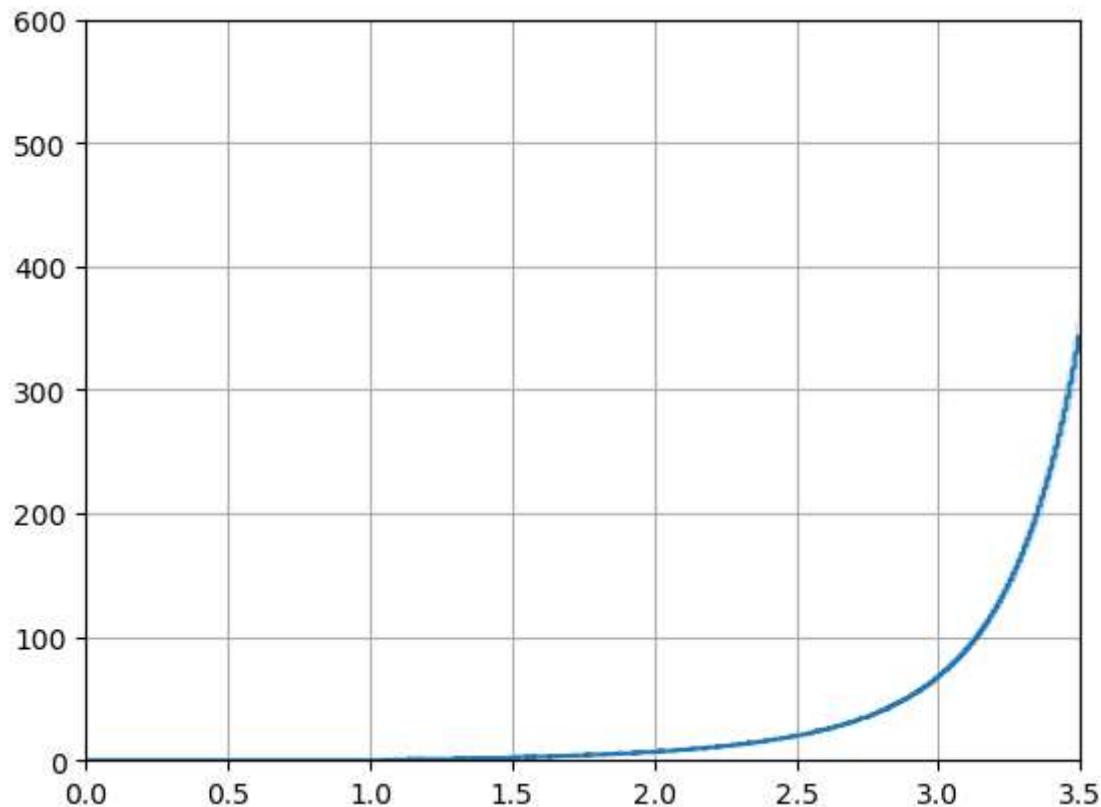
```
In [152...]: CF_08_Dry_IT = first_compaction(CF_08_Dry_IT_001, CF_08_Dry_IT_003, CF_08_Dry_IT_004)
```



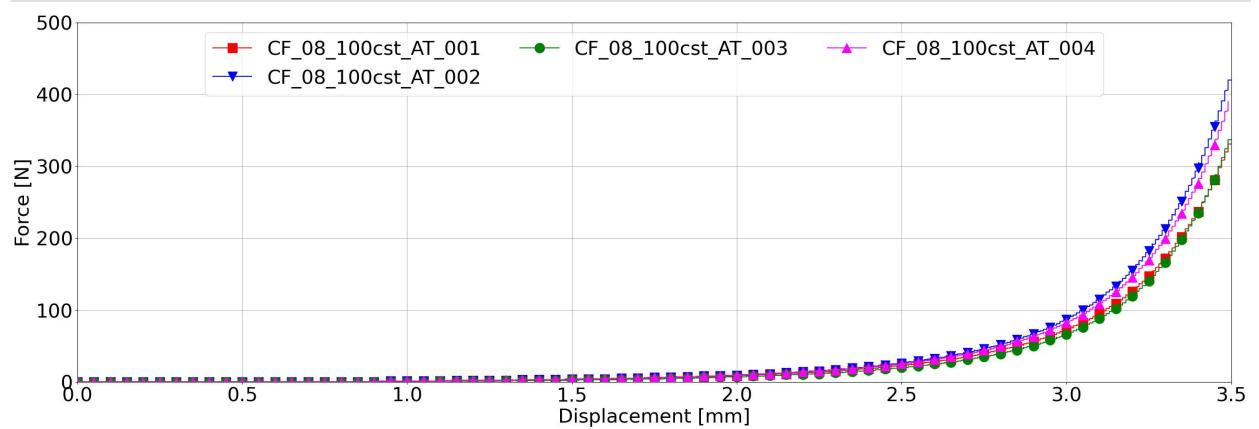
```
In [153...]: plot_force_displacement_single(CF_08_20cst_AT_001, 'CF_08_20cst_AT_001', CF_08_20cst_A
```



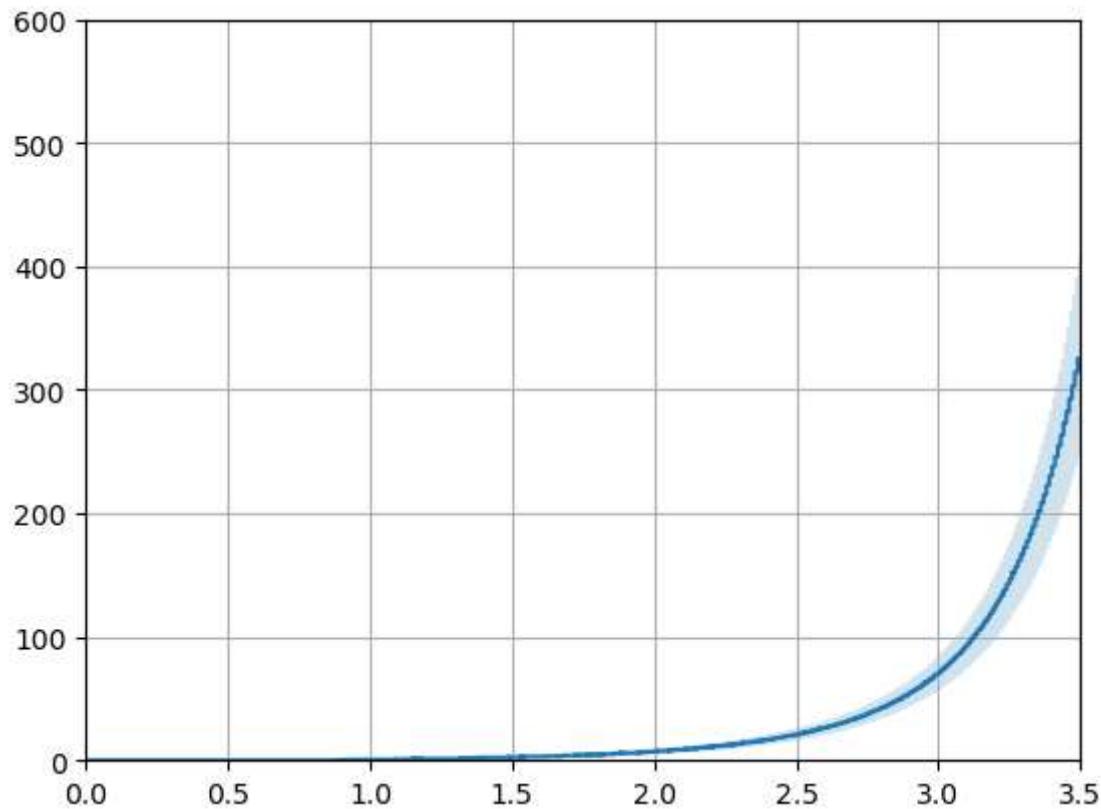
```
In [154...]: CF_08_20cst_AT = first_compaction(CF_08_20cst_AT_001, CF_08_20cst_AT_002, CF_08_20cst_A
```



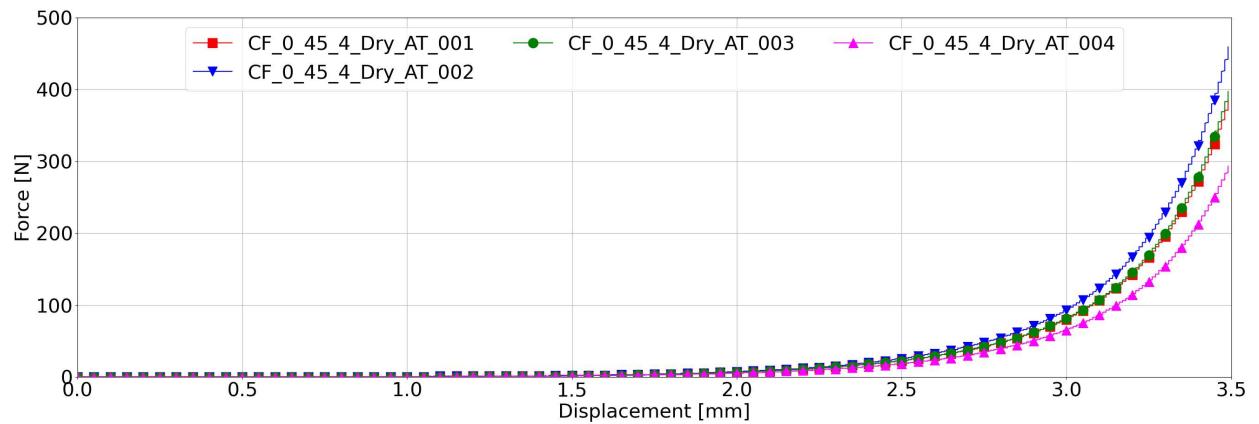
```
In [138...]: plot_force_displacement_single(CF_08_100cst_AT_001, 'CF_08_100cst_AT_001', CF_08_100cs
```



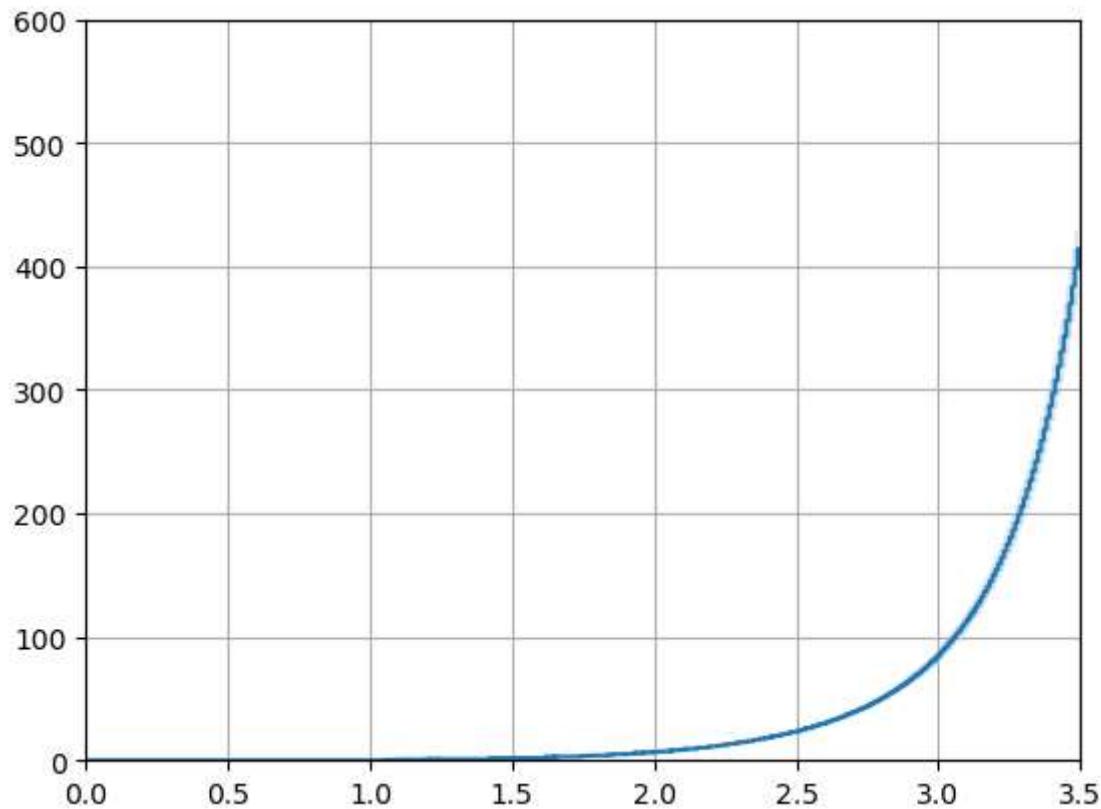
```
In [155...]: CF_08_100cst_AT = first_compaction(CF_08_100cst_AT_001, CF_08_100cst_AT_002, CF_08_Dry
```



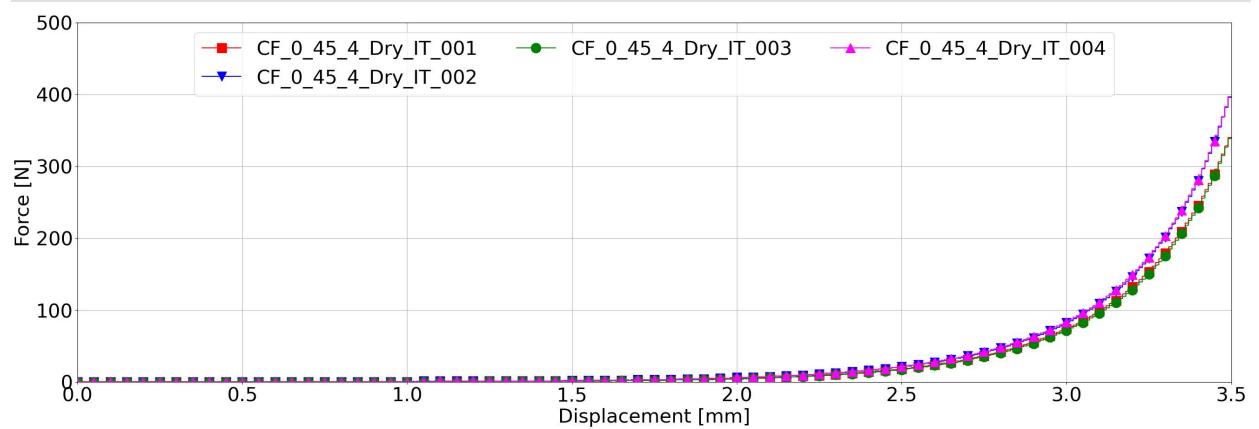
```
In [140...]: plot_force_displacement_single(CF_0_45_4_Dry_AT_001, 'CF_0_45_4_Dry_AT_001', CF_0_45_4
```



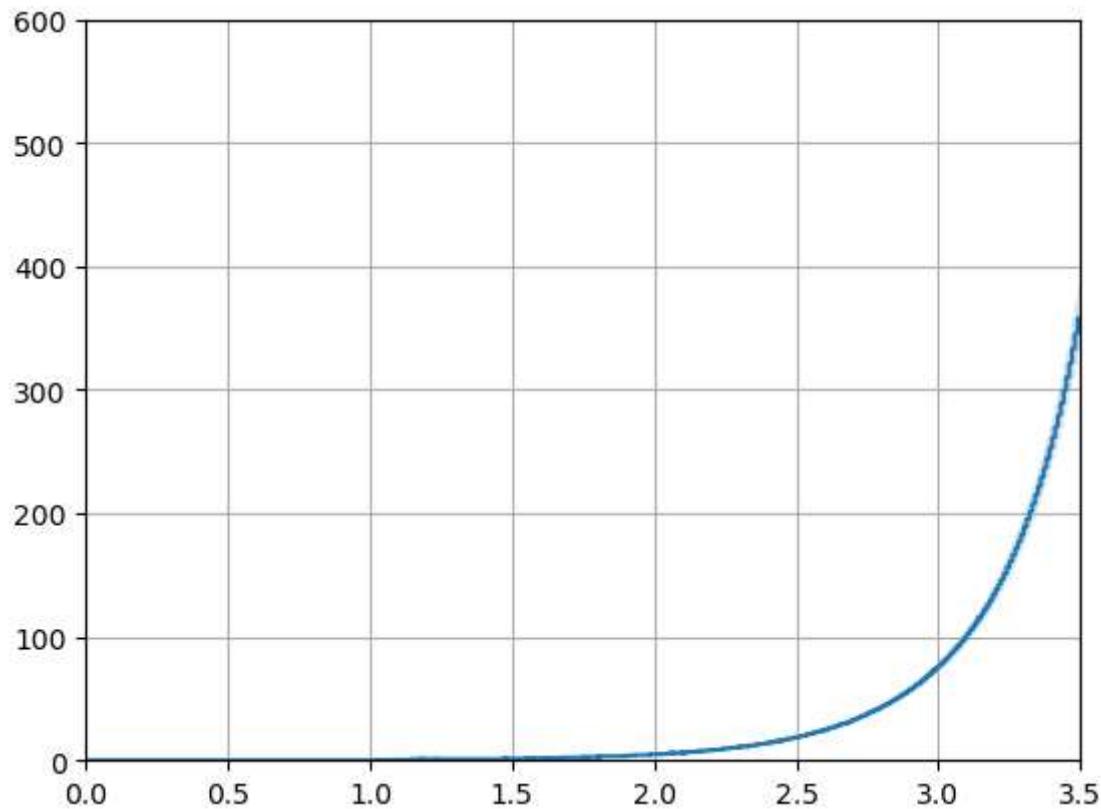
```
In [156...]: CF_0_45_4_Dry_AT = first_compaction(CF_0_45_4_Dry_AT_001, CF_0_45_4_Dry_AT_002, CF_0_4
```



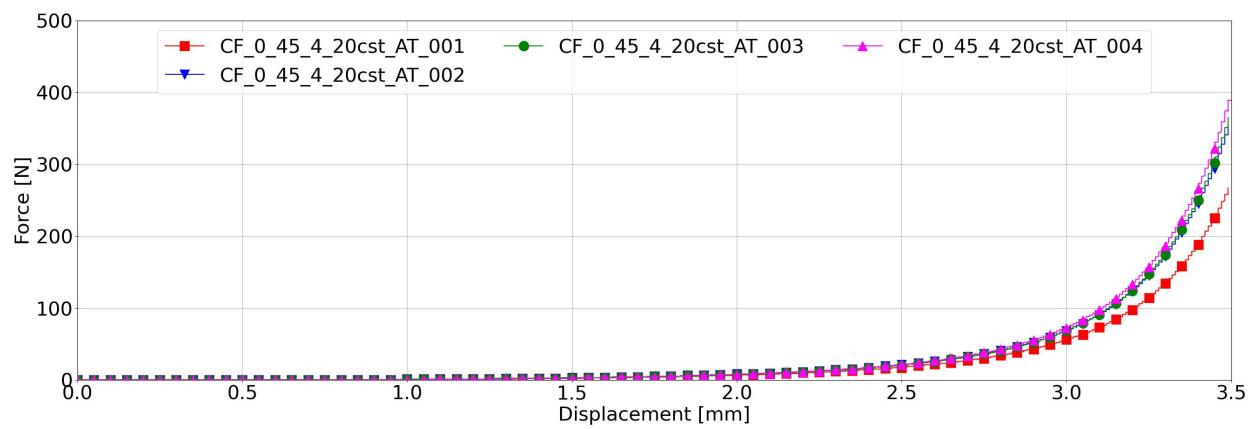
```
In [141]: plot_force_displacement_single(CF_0_45_4_Dry_IT_001, 'CF_0_45_4_Dry_IT_001', CF_0_45_4
```



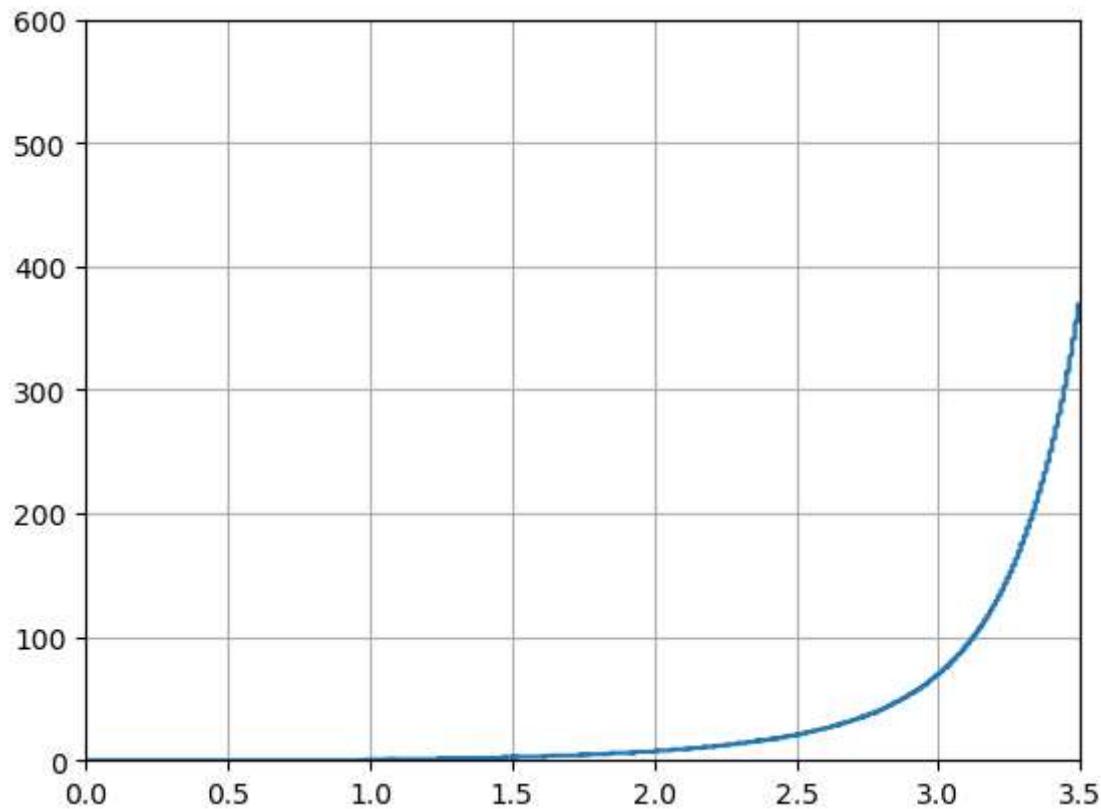
```
In [157]: CF_0_45_4_Dry_IT = first_compaction(CF_0_45_4_Dry_IT_001, CF_0_45_4_Dry_IT_002, CF_0_4
```



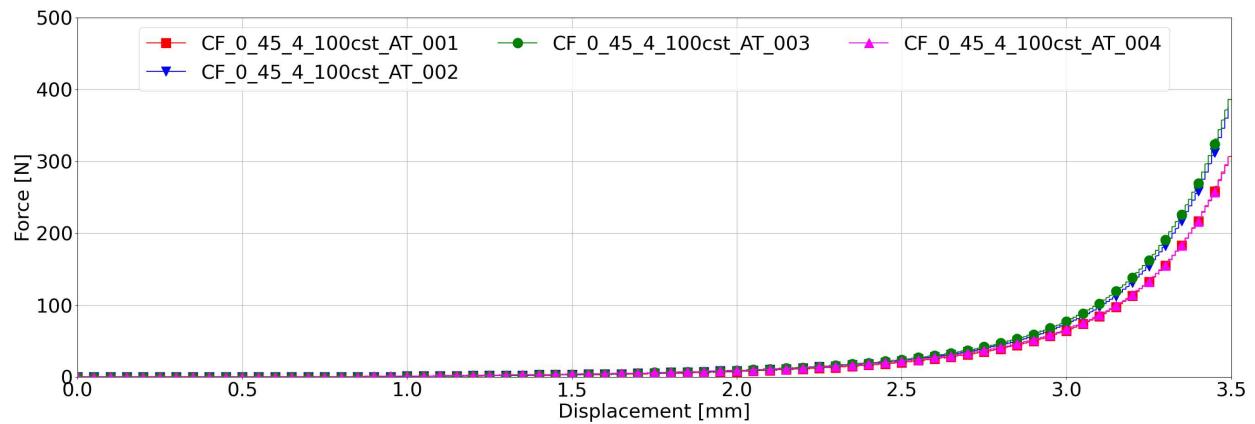
```
In [142...]: plot_force_displacement_single(CF_0_45_4_20cst_AT_001, 'CF_0_45_4_20cst_AT_001', CF_0_
```



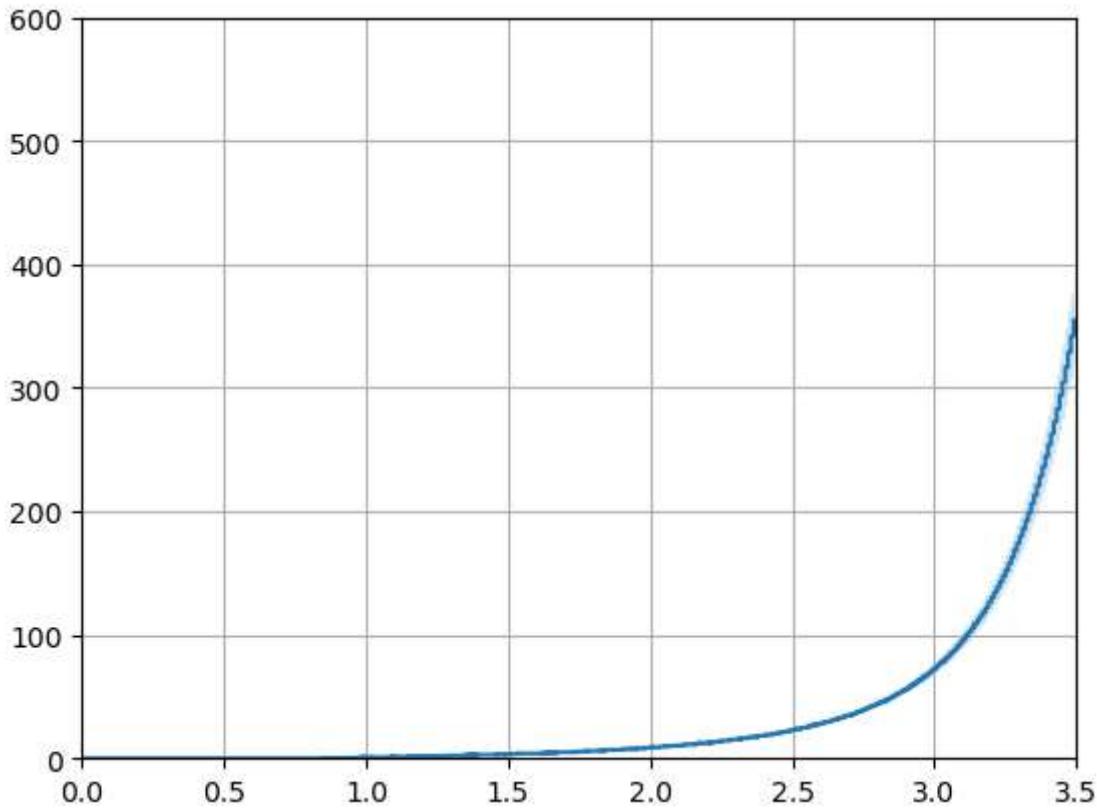
```
In [158...]: CF_0_45_4_20cst_AT = first_compaction(CF_0_45_4_20cst_AT_002, CF_0_45_4_20cst_AT_003,
```



```
In [143...]: plot_force_displacement_single(CF_0_45_4_100cst_AT_001, 'CF_0_45_4_100cst_AT_001', CF_
```



```
In [159...]: CF_0_45_4_100cst_AT = first_compaction(CF_0_45_4_100cst_AT_002, CF_0_45_4_100cst_AT_00
```



```
In [162]: def plot_comparison(specimen_1, label1, specimen_2, label2, specimen_3 = {}, label3 = {}):

    # plot
    fig, ax1 = plt.subplots(figsize = (25, 10))
    ax1.set_ylabel('Force [N]', fontsize=30)
    ax1.set_xlabel('Displacement [mm]', fontsize=30)
    plt.xticks(fontsize=30)
    plt.yticks(fontsize=30)

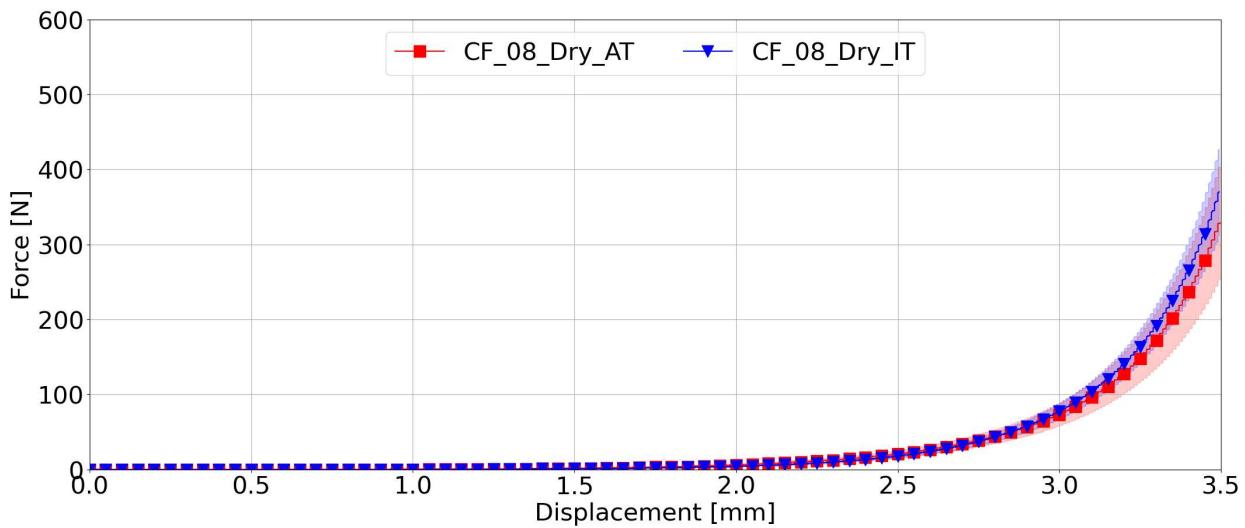
    # specimen 1
    ax1.plot(specimen_1['Displacement-mean'], specimen_1['Force-mean'], color='red',
             ax1.fill_between(specimen_1['Displacement-mean'], specimen_1['Force-std'][0], speci

    # specimen 2
    ax1.plot(specimen_2['Displacement-mean'], specimen_2['Force-mean'], color='blue',
             ax1.fill_between(specimen_2['Displacement-mean'], specimen_2['Force-std'][0], speci

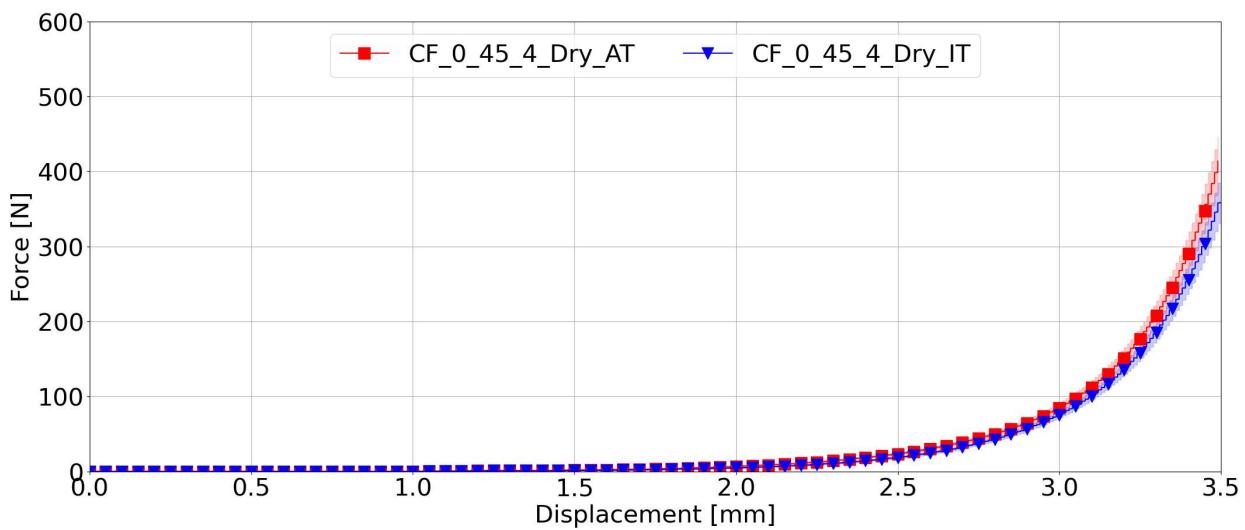
    # specimen 3
    if (bool(specimen_3)):
        ax1.plot(specimen_3['Displacement-mean'], specimen_3['Force-mean'], color='green',
                 ax1.fill_between(specimen_3['Displacement-mean'], specimen_3['Force-std'][0], speci

    ax1.grid()
    ax1.set_xlim(0,3.5)
    ax1.set_ylim(0,600)
    ax1.legend(fontsize="30", loc='upper center', ncol=3)
```

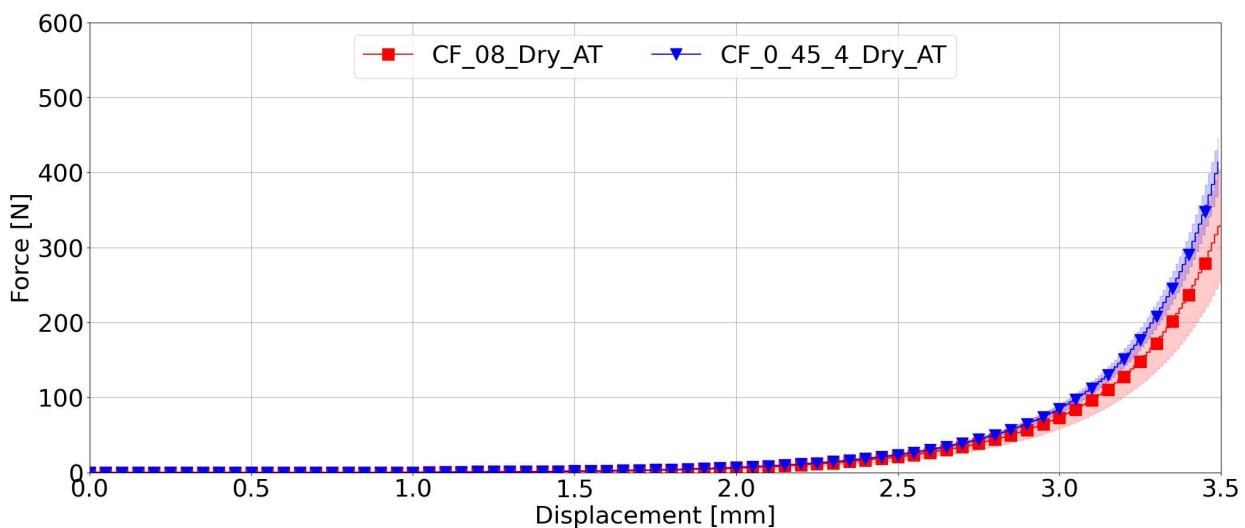
```
In [163]: plot_comparison(CF_08_Dry_AT, 'CF_08_Dry_AT', CF_08_Dry_IT, 'CF_08_Dry_IT')
```



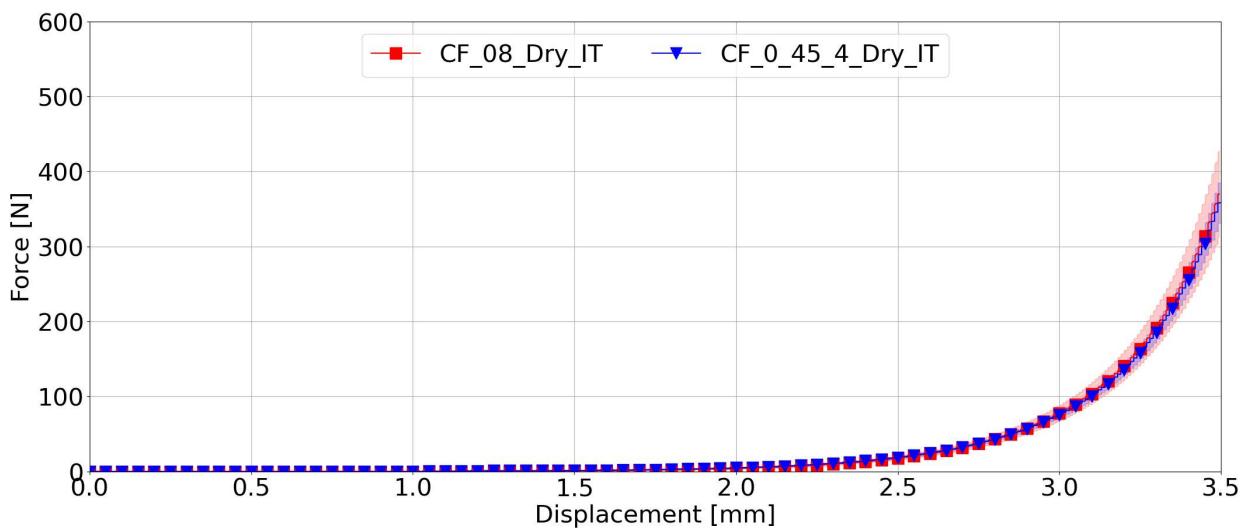
```
In [164...]: plot_comparison(CF_08_Dry_AT, 'CF_08_Dry_AT', CF_08_Dry_IT, 'CF_08_Dry_IT')
```



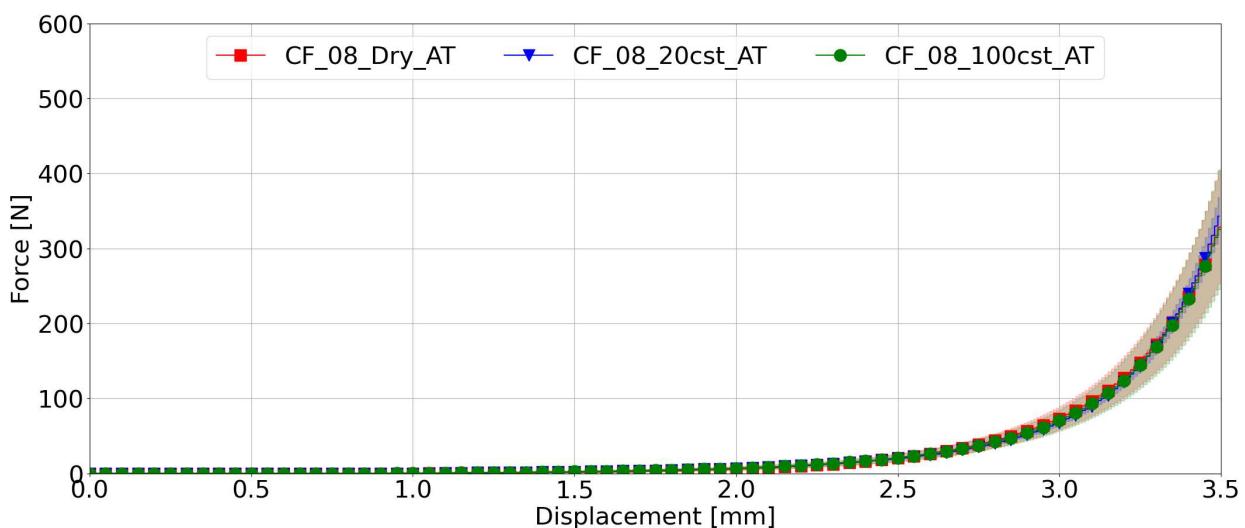
```
In [167...]: plot_comparison(CF_08_Dry_AT, 'CF_08_Dry_AT', CF_045_4_Dry_AT, 'CF_045_4_Dry_AT')
```



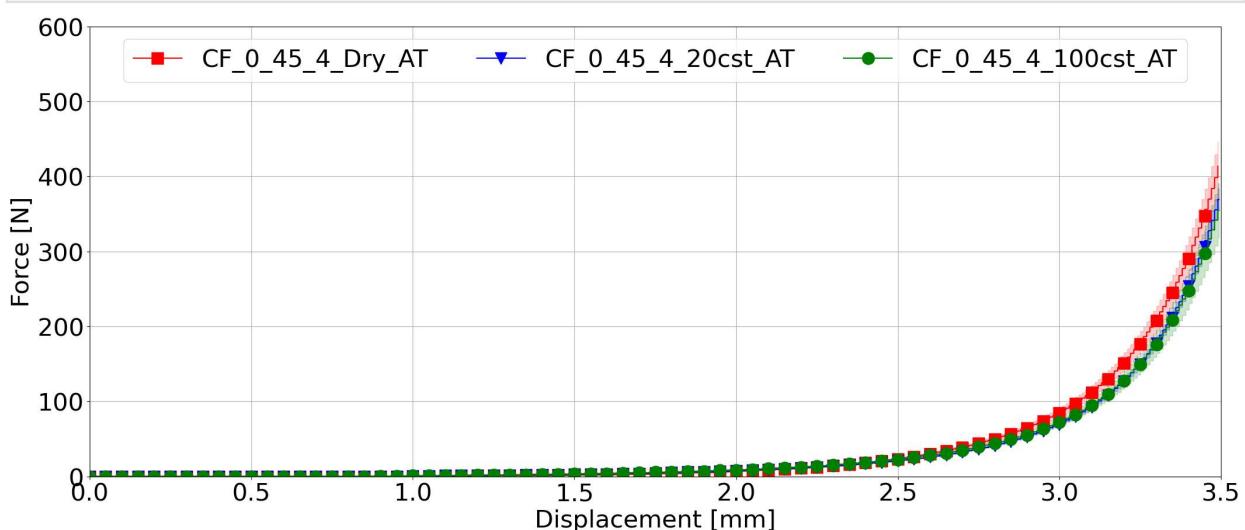
```
In [168...]: plot_comparison(CF_08_Dry_IT, 'CF_08_Dry_IT', CF_045_4_Dry_IT, 'CF_045_4_Dry_IT')
```



```
In [165]: plot_comparison(CF_08_Dry_IT, 'CF_08_Dry_AT', CF_08_20cst_AT, 'CF_08_20cst_AT', CF_08_100cst_AT)
```



```
In [166]: plot_comparison(CF_0_45_4_Dry_AT, 'CF_0_45_4_Dry_AT', CF_0_45_4_20cst_AT, 'CF_0_45_4_20cst_AT', CF_0_45_4_100cst_AT)
```



```
In [40]: def compaction(specimen1, specimen2, specimen3 = pd.DataFrame({'A' : []}), specimen4 = pd.DataFrame({'A' : []})):
    array = np.linspace(0,specimen1['Kraft_N'].size - 1,specimen1['Kraft_N'].size)
    specimen1['index'] = array
```

```

array = np.linspace(0,specimen2['Kraft_N'].size - 1,specimen2['Kraft_N'].size)
specimen2['index'] = array

if (not specimen3.empty):
    array = np.linspace(0,specimen3['Kraft_N'].size - 1,specimen3['Kraft_N'].size)
    specimen3['index'] = array

if (not specimen4.empty):
    array = np.linspace(0,specimen4['Kraft_N'].size - 1,specimen4['Kraft_N'].size)
    specimen4['index'] = array

vector_1 = (specimen1['Kraft_N'] != 0) & (specimen1['index'] < specimen1['Kraft_N'])
vector_2 = (specimen2['Kraft_N'] != 0) & (specimen2['index'] < specimen2['Kraft_N'])

if (not specimen3.empty) & (specimen4.empty):
    numRows = min(specimen1.iloc[:, 0].size, specimen2.iloc[:, 0].size, specimen3.size)
    vector_3 = (specimen3['Kraft_N'] != 0) & (specimen3['index'] < specimen3['Kraft_N'])
    vectors = [vector_1, vector_2, vector_3]
    numCols = 3
elif (specimen3.empty) & (not specimen4.empty):
    numRows = min(specimen1.iloc[:, 0].size, specimen2.iloc[:, 0].size, specimen4.size)
    vector_4 = (specimen4['Kraft_N'] != 0) & (specimen4['index'] < specimen4['Kraft_N'])
    vectors = [vector_1, vector_2, vector_4]
    numCols = 3
elif (not specimen3.empty) & (not specimen4.empty):
    numRows = min(specimen1.iloc[:, 0].size, specimen2.iloc[:, 0].size, specimen3.size, specimen4.size)
    vector_3 = (specimen3['Kraft_N'] != 0) & (specimen3['index'] < specimen3['Kraft_N'])
    vector_4 = (specimen4['Kraft_N'] != 0) & (specimen4['index'] < specimen4['Kraft_N'])
    vectors = [vector_1, vector_2, vector_3, vector_4]
    numCols = 4
elif (specimen3.empty) & (specimen4.empty):
    numRows = min(specimen1.iloc[:, 0].size, specimen2.iloc[:, 0].size)
    vectors = [vector_1, vector_2]
    numCols = 2

vector = vector_1

for i in range(numCols):
    if sum(vectors[i]) > sum(vector):
        vector = vectors[i]

# Force dataframe
force = pd.DataFrame(index=range(numRows),columns=range(numCols))
force.iloc[:, 0] = specimen1.iloc[0:numRows,1]
force.iloc[:, 1] = specimen2.iloc[0:numRows,1]
if (not specimen3.empty):
    force.iloc[:, 2] = specimen3.iloc[0:numRows,1]
if (not specimen4.empty):
    force.iloc[:, 3] = specimen4.iloc[0:numRows,1]

## mean value and std
force_mean = np.mean(force[vector], axis = 1)
force_std = np.std(force[vector], axis = 1)
force_std_range = (force_mean - force_std, force_mean + force_std)#

# Displacement dataframe
displacement = pd.DataFrame(index=range(numRows),columns=range(numCols))
displacement.iloc[:, 0] = specimen1.iloc[0:numRows,2]

```

```

displacement.iloc[:, 1] = specimen2.iloc[0:numRows,2]
initial_value = [specimen1['Weg_mm'][vector_1].iloc[0], specimen2['Weg_mm'][vector_1].iloc[0]]
if (not specimen3.empty):
    displacement.iloc[:, 2] = specimen3.iloc[0:numRows,2]
    initial_value.append(specimen3['Weg_mm'][vector_3].iloc[0])
if (not specimen4.empty):
    displacement.iloc[:, 3] = specimen4.iloc[0:numRows,2]
    initial_value.append(specimen4['Weg_mm'][vector_4].iloc[0])

initial_value = np.mean(initial_value)
## mean value and std

displacement_mean = np.absolute(initial_value - np.mean(displacement[vector], axis = 1))
#displacement_std = np.std(displacement[vector], axis = 1)
#displacement_std_range = (displacement_mean - displacement_std, displacement_mean + displacement_std)

dictionary = {'Force-mean': force_mean, 'Force-std': force_std_range, 'Displacement-mean': displacement_mean, 'Displacement-std': displacement_std_range}

plt.plot(displacement_mean, force_mean)
plt.fill_between(dictionary['Displacement-mean'], dictionary['Force-std'][0], dictionary['Force-std'][1], alpha = 0.2)

return dictionary

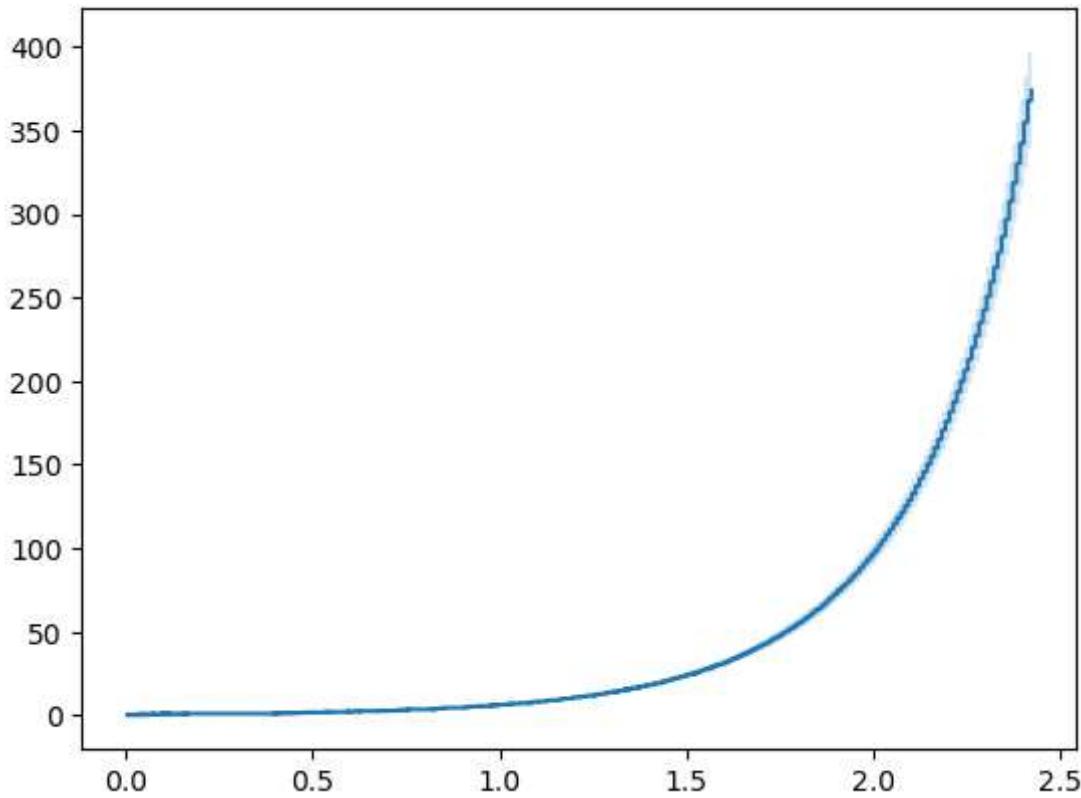
```

In [79]: CF_0_45_4_Dry_IT = compaction(CF_0_45_4_Dry_IT_001,CF_0_45_4_Dry_IT_002, CF_0_45_4_Dry_IT_003)

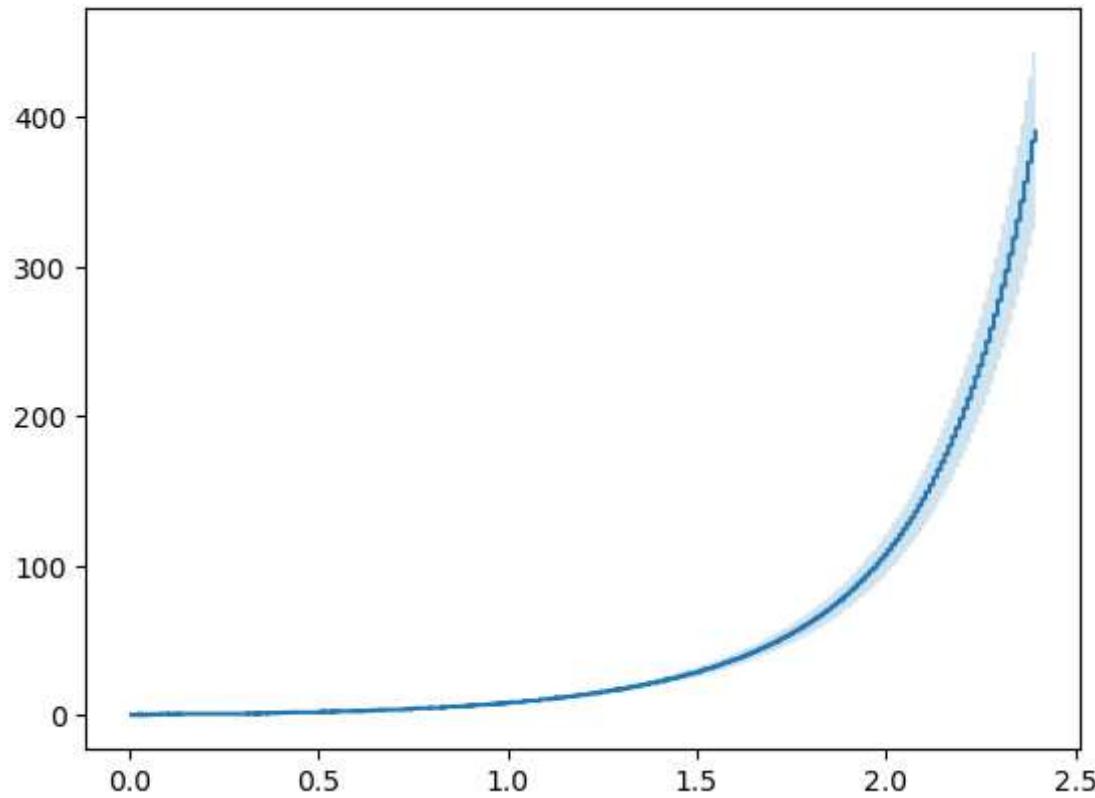
```

C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:58: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  force_mean = np.mean(force[vector], axis = 1)
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:59: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  force_std = np.std(force[vector], axis = 1)
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:77: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  displacement_mean = np.absolute(initial_value - np.mean(displacement[vector], axis = 1))

```

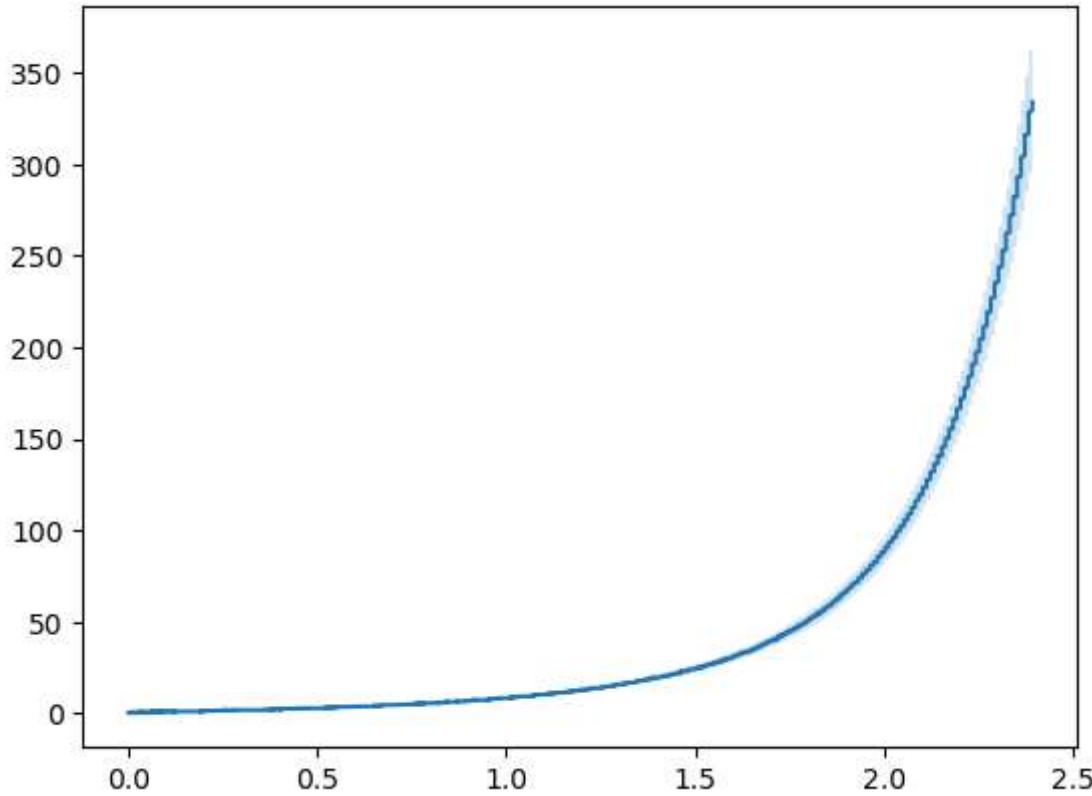


```
In [90]: CF_0_45_4_Dry_AT = compaction(CF_0_45_4_Dry_AT_001,CF_0_45_4_Dry_AT_002, CF_0_45_4_Dry  
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:58: UserWarning:  
g: Boolean Series key will be reindexed to match DataFrame index.  
    force_mean = np.mean(force[vector], axis = 1)  
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:59: UserWarning:  
g: Boolean Series key will be reindexed to match DataFrame index.  
    force_std = np.std(force[vector], axis = 1)  
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:77: UserWarning:  
g: Boolean Series key will be reindexed to match DataFrame index.  
    displacement_mean = np.absolute(initial_value - np.mean(displacement[vector], axis  
= 1))
```



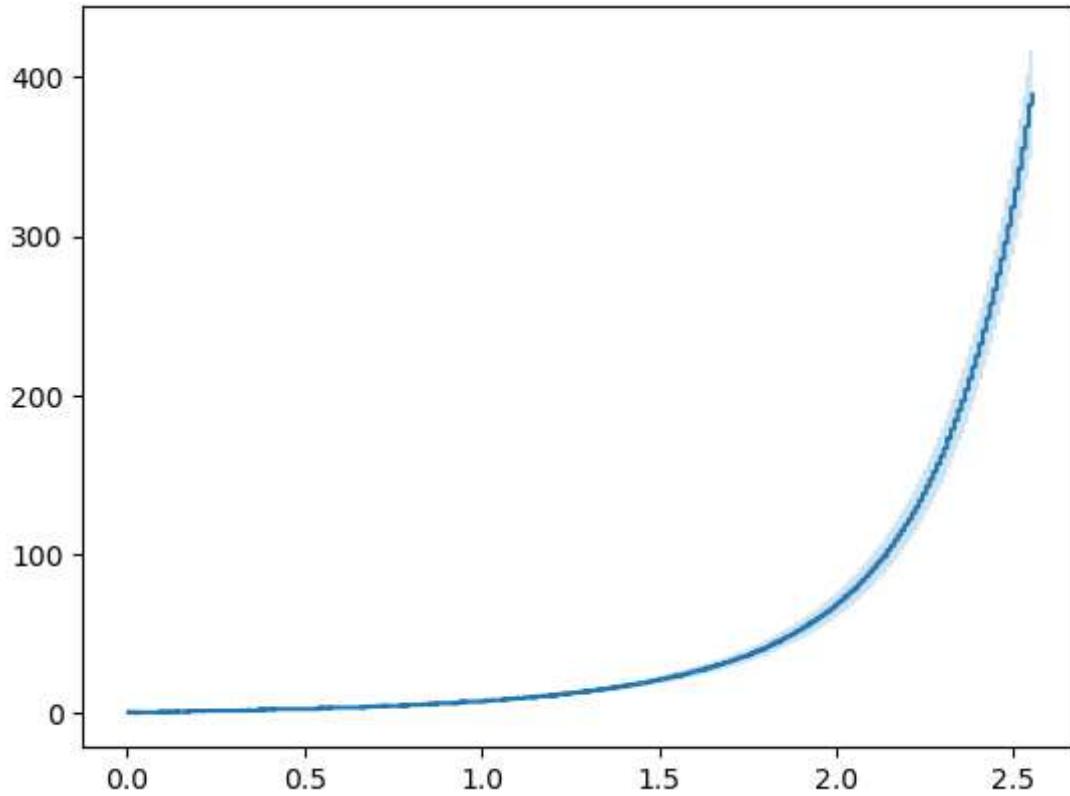
```
In [43]: CF_08_20cst_AT = compaction(CF_08_20cst_AT_001, CF_08_20cst_AT_002, CF_08_20cst_AT_003)
```

```
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:58: UserWarning:  
  Boolean Series key will be reindexed to match DataFrame index.  
    force_mean = np.mean(force[vector], axis = 1)  
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:59: UserWarning:  
  Boolean Series key will be reindexed to match DataFrame index.  
    force_std = np.std(force[vector], axis = 1)  
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:77: UserWarning:  
  Boolean Series key will be reindexed to match DataFrame index.  
    displacement_mean = np.absolute(initial_value - np.mean(displacement[vector], axis  
= 1))
```

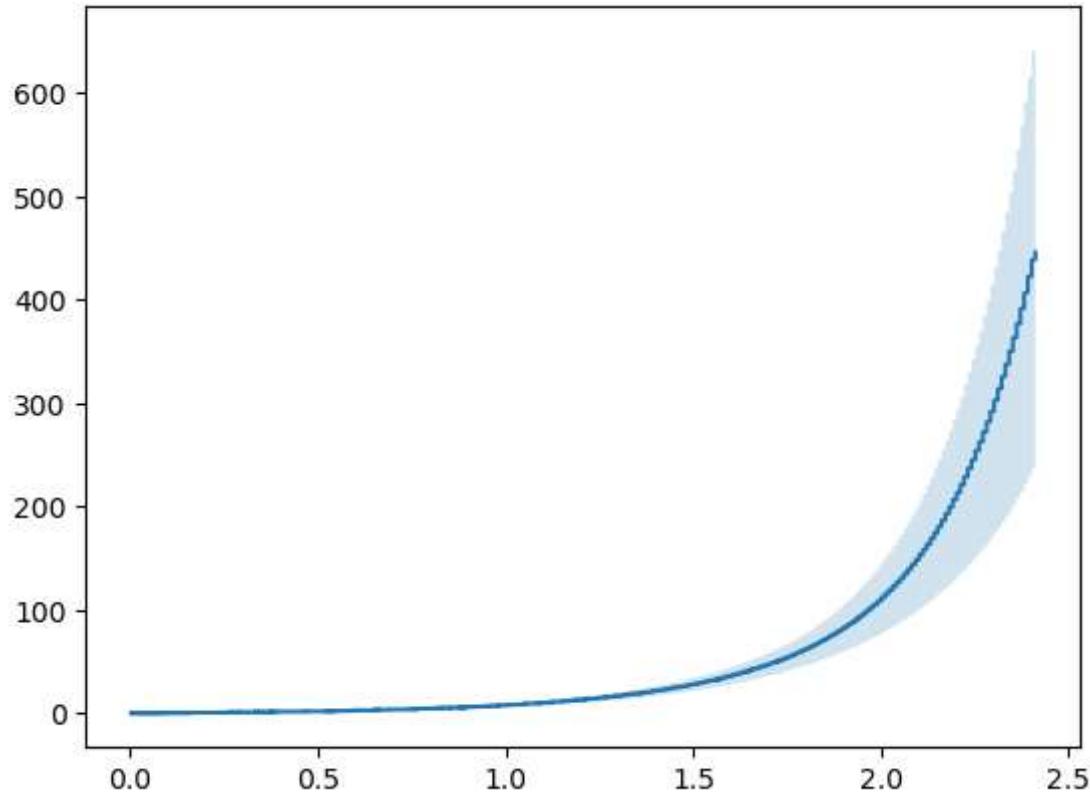


```
In [45]: CF_08_100cst_AT = compaction(CF_08_100cst_AT_002, CF_08_100cst_AT_003, CF_08_100cst_AT_004)

C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:58: UserWarning:
  Boolean Series key will be reindexed to match DataFrame index.
    force_mean = np.mean(force[vector], axis = 1)
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:59: UserWarning:
  Boolean Series key will be reindexed to match DataFrame index.
    force_std = np.std(force[vector], axis = 1)
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:77: UserWarning:
  Boolean Series key will be reindexed to match DataFrame index.
    displacement_mean = np.absolute(initial_value - np.mean(displacement[vector], axis = 1))
```

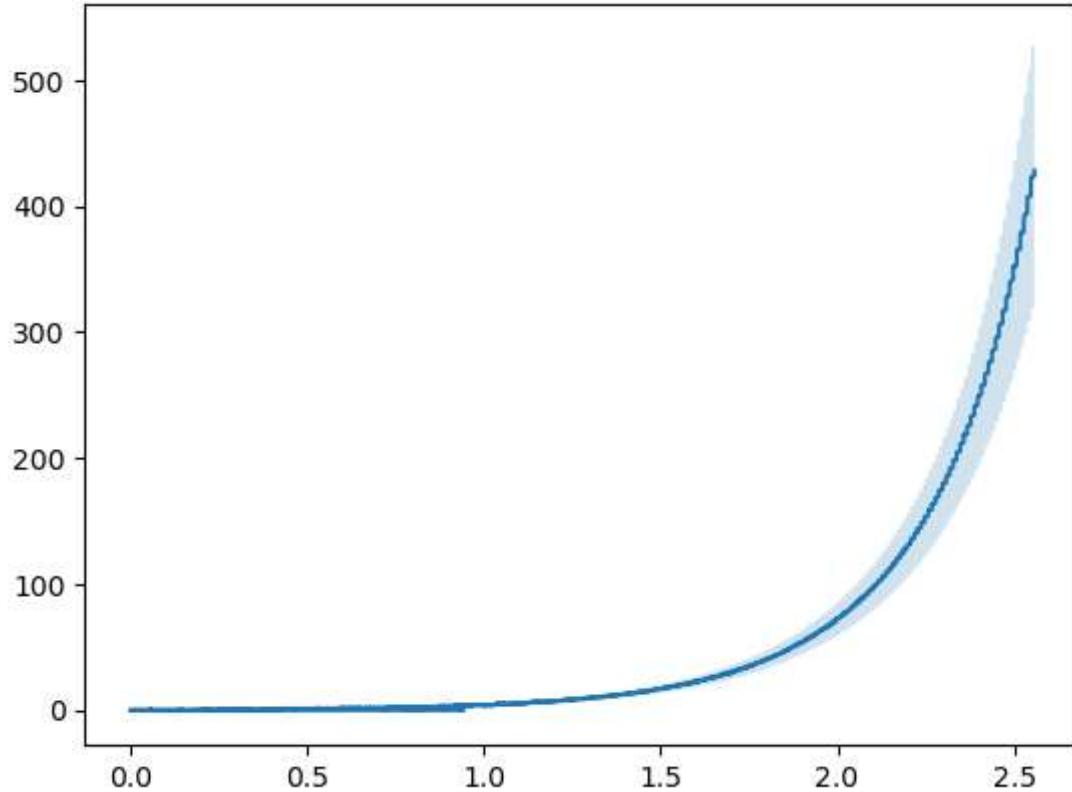


```
In [89]: CF_08_Dry_AT = compaction(CF_08_Dry_AT_001, CF_08_Dry_AT_002, CF_08_Dry_AT_003, CF_08_
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:58: UserWarning:
g: Boolean Series key will be reindexed to match DataFrame index.
    force_mean = np.mean(force[vector], axis = 1)
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:59: UserWarning:
g: Boolean Series key will be reindexed to match DataFrame index.
    force_std = np.std(force[vector], axis = 1)
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:77: UserWarning:
g: Boolean Series key will be reindexed to match DataFrame index.
    displacement_mean = np.absolute(initial_value - np.mean(displacement[vector], axis
= 1))
```



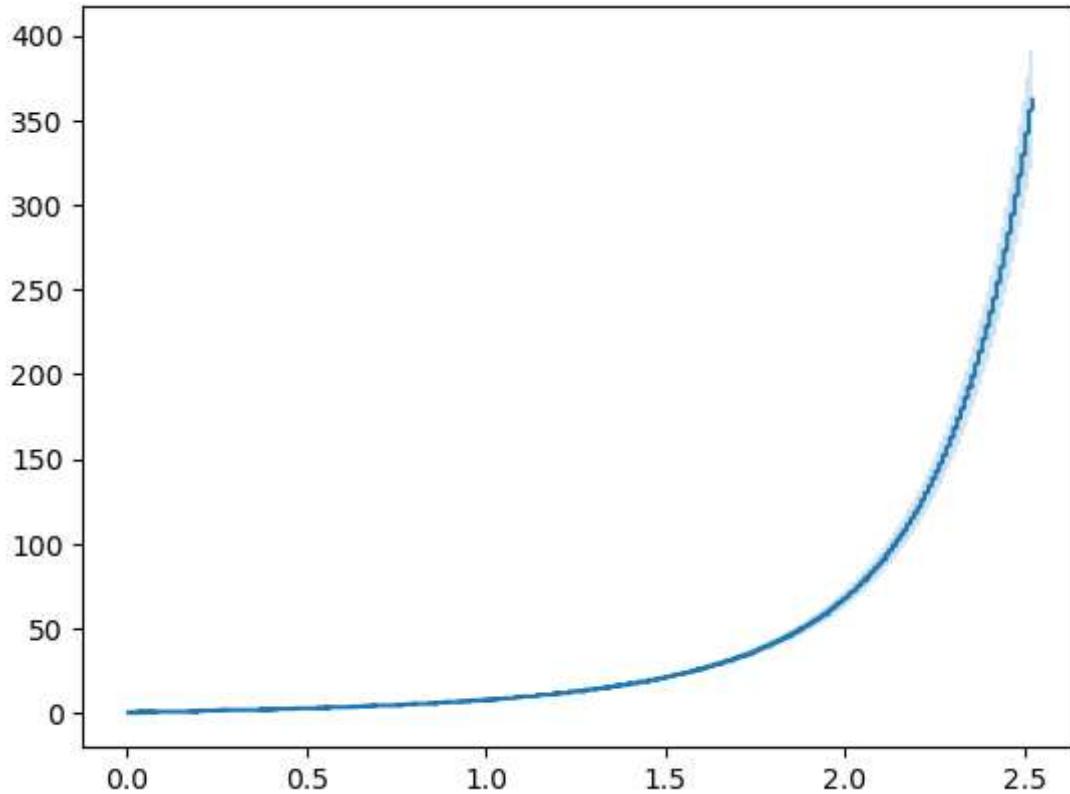
```
In [88]: CF_08_Dry_IT = compaction(CF_08_Dry_IT_002, CF_08_Dry_IT_003, CF_08_Dry_IT_004, CF_08_
```

C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:58: UserWarning:
 Boolean Series key will be reindexed to match DataFrame index.
 force_mean = np.mean(force[vector], axis = 1)
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:59: UserWarning:
 Boolean Series key will be reindexed to match DataFrame index.
 force_std = np.std(force[vector], axis = 1)
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:77: UserWarning:
 Boolean Series key will be reindexed to match DataFrame index.
 displacement_mean = np.absolute(initial_value - np.mean(displacement[vector], axis
= 1))



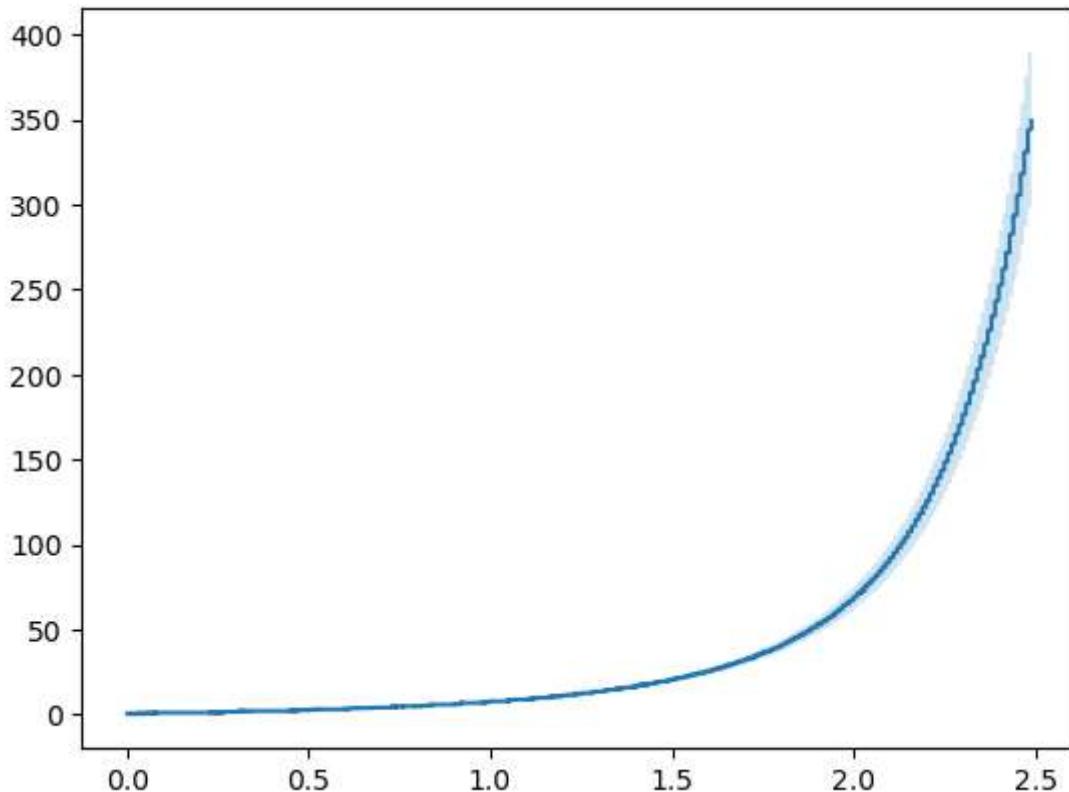
```
In [87]: CF_0_45_4_100cst_AT = compaction(CF_0_45_4_100cst_AT_001, CF_0_45_4_100cst_AT_002, CF_0_45_4_100cst_AT_003)

C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:58: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
    force_mean = np.mean(force[vector], axis = 1)
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:59: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
    force_std = np.std(force[vector], axis = 1)
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:77: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
    displacement_mean = np.absolute(initial_value - np.mean(displacement[vector], axis = 1))
```



```
In [50]: CF_0_45_4_20cst_AT = compaction(CF_0_45_4_20cst_AT_001, CF_0_45_4_20cst_AT_002, CF_0_45_4_20cst_AT_003)

C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:58: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
    force_mean = np.mean(force[vector], axis = 1)
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:59: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
    force_std = np.std(force[vector], axis = 1)
C:\Users\Renan Portela\AppData\Local\Temp\ipykernel_3308\977365587.py:77: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
    displacement_mean = np.absolute(initial_value - np.mean(displacement[vector], axis = 1))
```



```
In [57]: fig, ax = plt.subplots(figsize = (30, 12))

ax.plot(CF_0_45_4_Dry_AT['Displacement-mean'], CF_0_45_4_Dry_AT['Force-mean'], marker='o', markevery=15)
ax.fill_between(CF_0_45_4_Dry_AT['Displacement-mean'], CF_0_45_4_Dry_AT['Force-std'][0],CF_0_45_4_Dry_AT['Force-std'][len(CF_0_45_4_Dry_AT['Displacement-mean'])-1], alpha=0.2)

ax.plot(CF_0_45_4_Dry_IT['Displacement-mean'], CF_0_45_4_Dry_IT['Force-mean'], marker='s', markevery=15)
ax.fill_between(CF_0_45_4_Dry_IT['Displacement-mean'], CF_0_45_4_Dry_IT['Force-std'][0],CF_0_45_4_Dry_IT['Force-std'][len(CF_0_45_4_Dry_IT['Displacement-mean'])-1], alpha=0.2)

ax.legend()

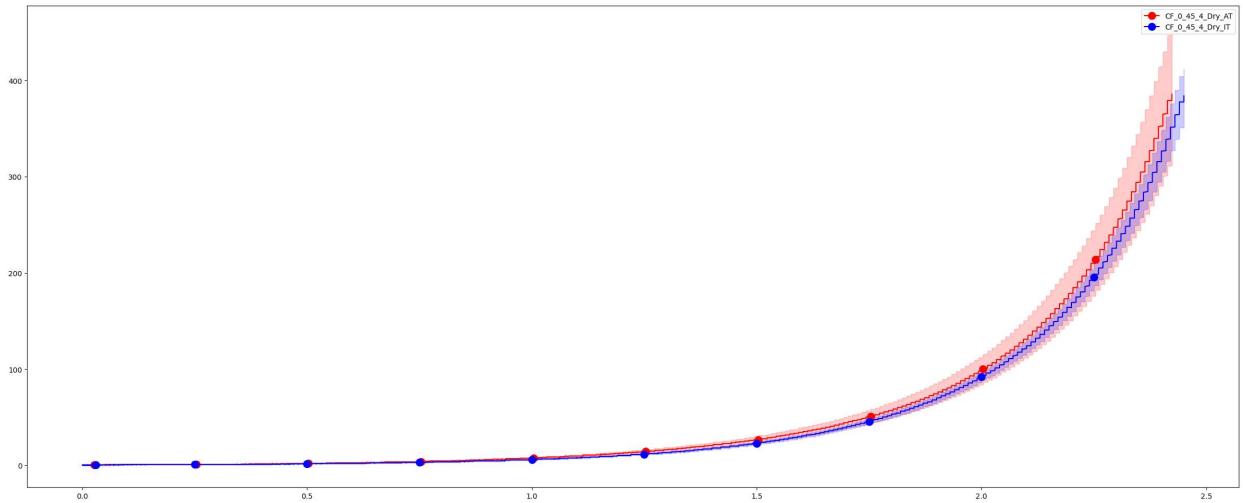
#ax1.plot(dictionary1['Time'], dictionary1['Force-mean'], marker = 'o', markevery = 15
#ax1.fill_between(dictionary1['Time'], dictionary1['Force-std'][0],dictionary1['Force-std'][len(dictionary1['Time'])-1], alpha=0.2)

#ax1.plot(dictionary2['Time'], dictionary2['Force-mean'], marker = 's', markevery = 15
#ax1.fill_between(dictionary2['Time'], dictionary2['Force-std'][0],dictionary2['Force-std'][len(dictionary2['Time'])-1], alpha=0.2)

#ax1.plot(dictionary3['Time'], dictionary3['Force-mean'], marker = '^', markevery = 15
#ax1.fill_between(dictionary3['Time'], dictionary3['Force-std'][0],dictionary3['Force-std'][len(dictionary3['Time'])-1], alpha=0.2)

plt.xticks(fontsize=30)
plt.yticks(fontsize=30)
ax2 = ax1.twinx()

Out[57]: <matplotlib.legend.Legend at 0x25bb8f30400>
```



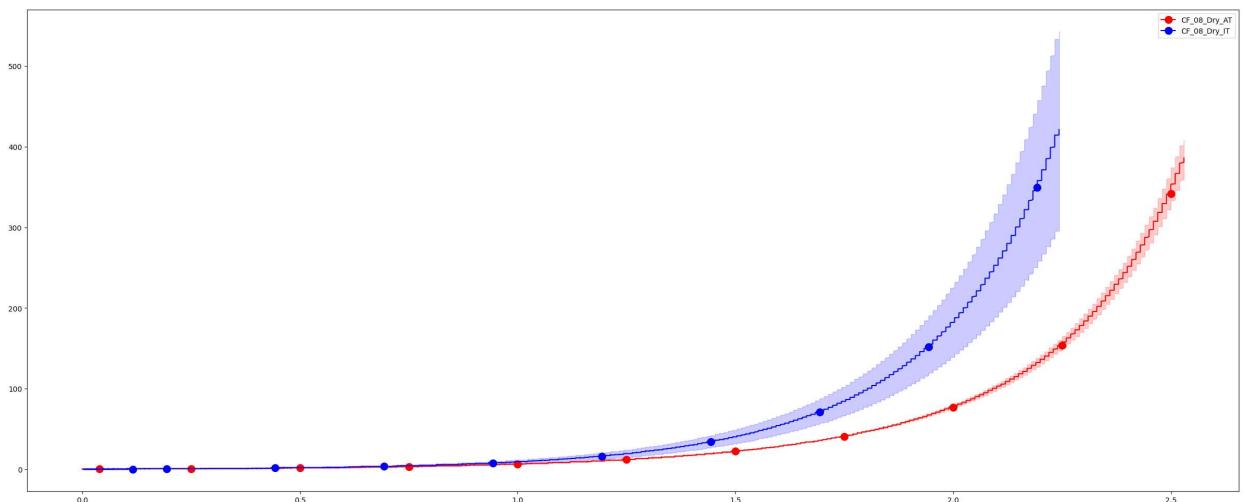
```
In [56]: fig, ax = plt.subplots(figsize = (30, 12))

ax.plot(CF_08_Dry_AT['Displacement-mean'], CF_08_Dry_AT['Force-mean'], marker = 'o', n
ax.fill_between(CF_08_Dry_AT['Displacement-mean'], CF_08_Dry_AT['Force-std'][0], CF_08

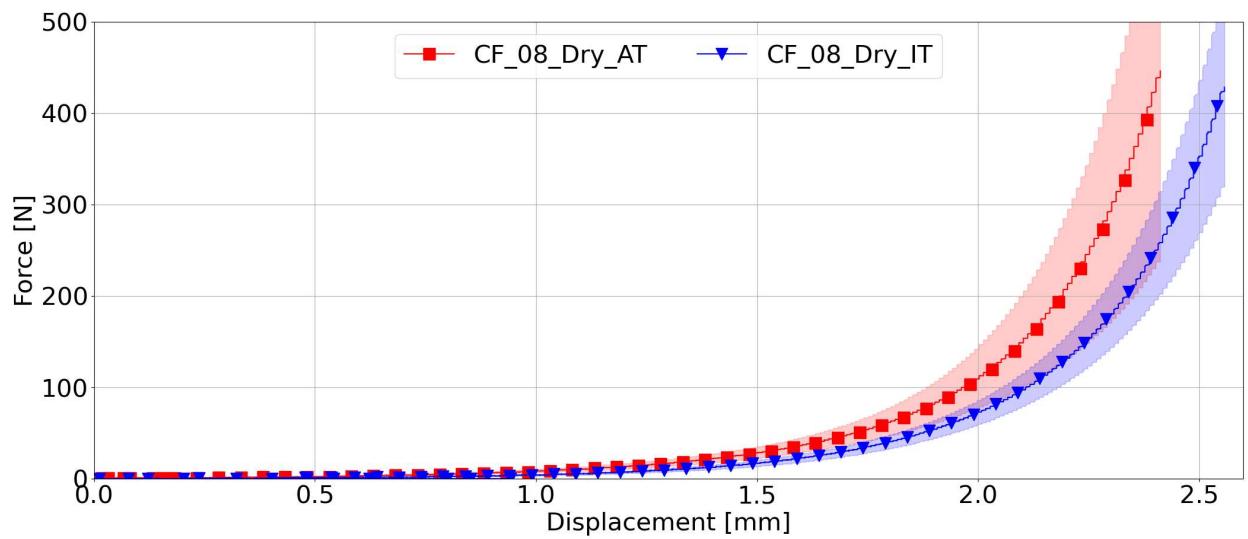
ax.plot(CF_08_Dry_IT['Displacement-mean'], CF_08_Dry_IT['Force-mean'], marker = 'o', n
ax.fill_between(CF_08_Dry_IT['Displacement-mean'], CF_08_Dry_IT['Force-std'][0], CF_08

ax.legend()
```

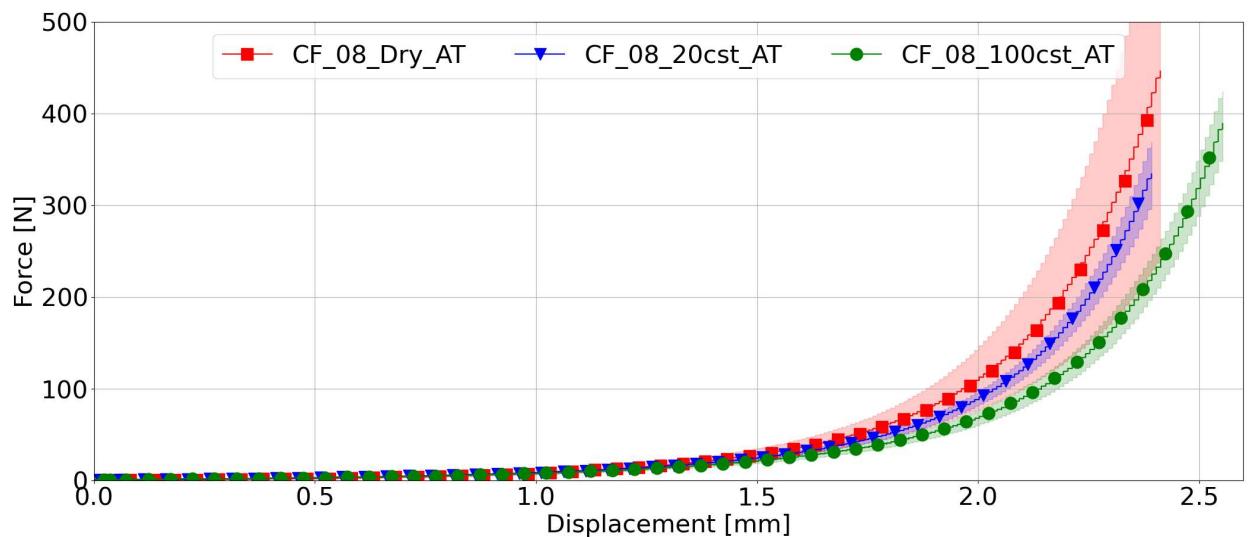
```
Out[56]: <matplotlib.legend.Legend at 0x25bb4704580>
```



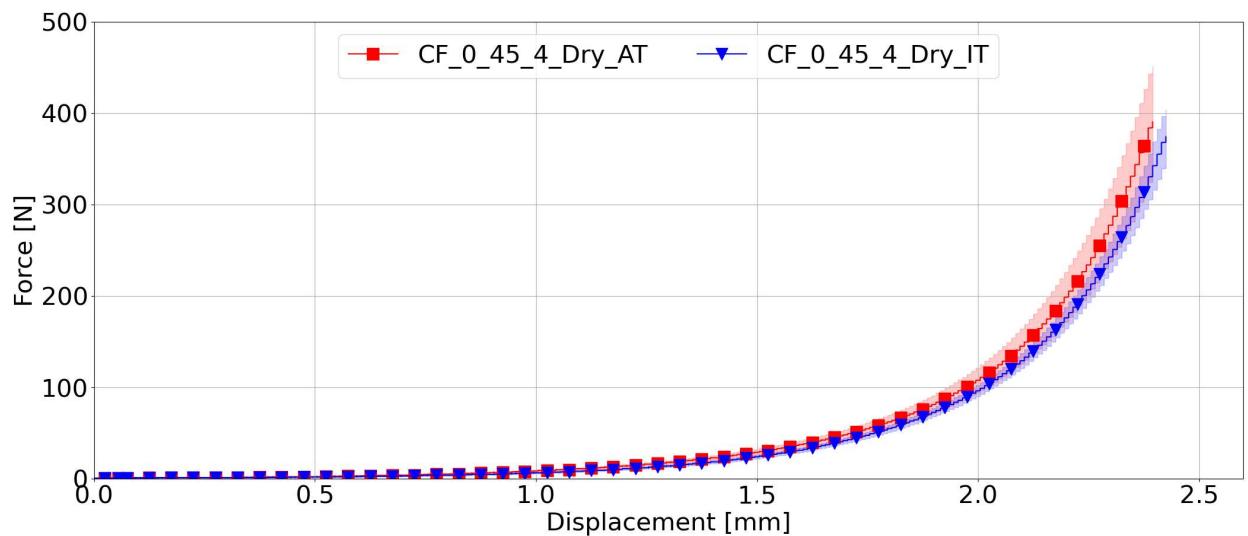
```
In [95]: plot_comparision(CF_08_Dry_AT, 'CF_08_Dry_AT', CF_08_Dry_IT, 'CF_08_Dry_IT')
```



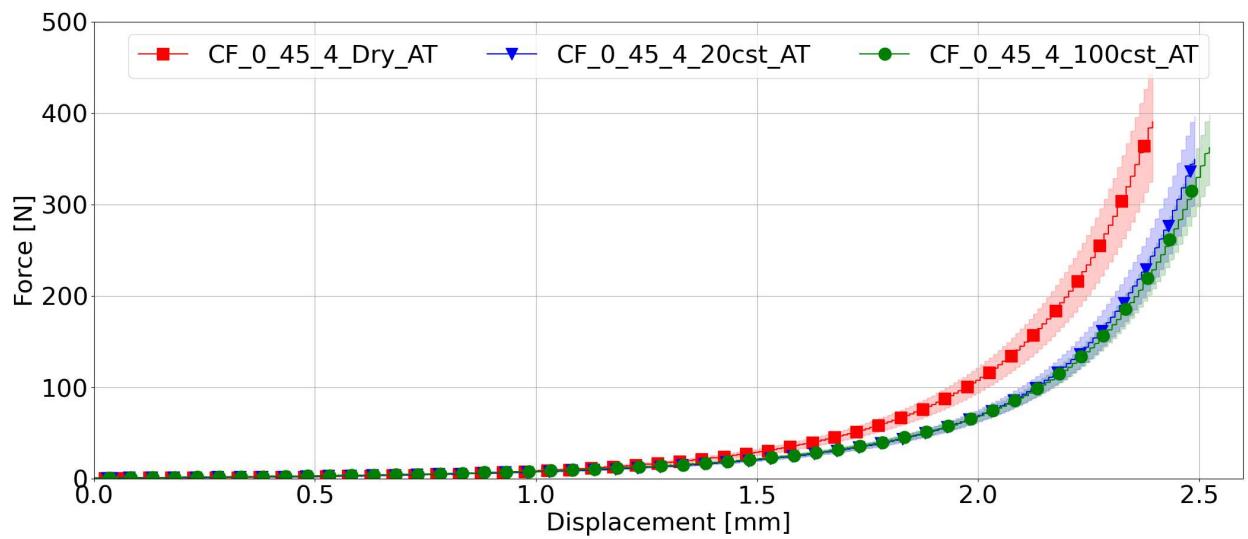
```
In [96]: plot_comparison(CF_08_Dry_AT, 'CF_08_Dry_AT', CF_08_20cst_AT, 'CF_08_20cst_AT', CF_08_100cst_AT, 'CF_08_100cst_AT')
```



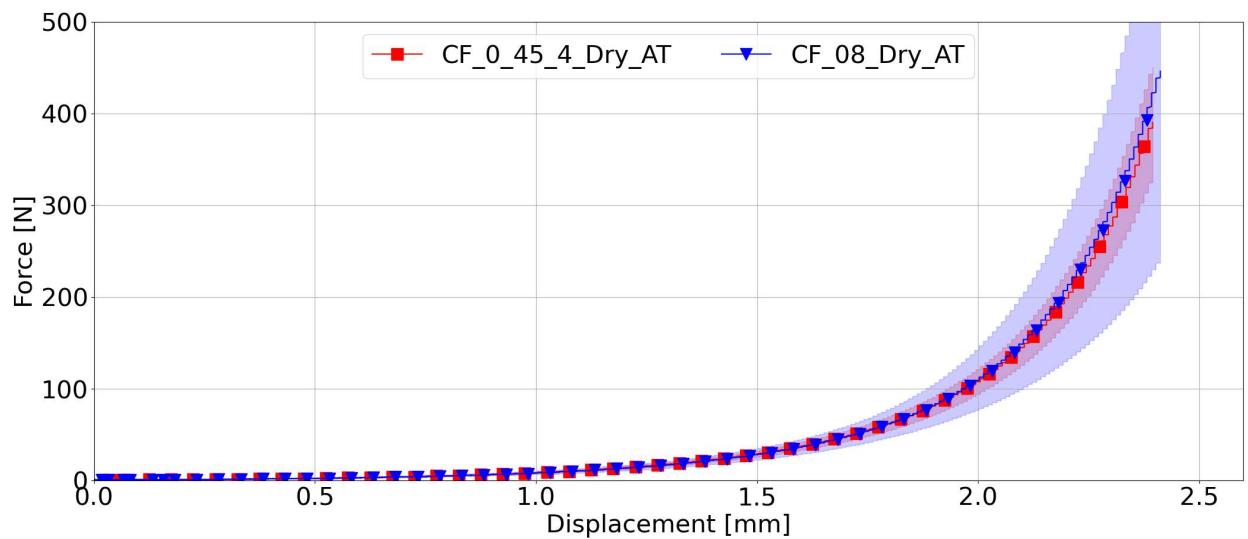
```
In [94]: plot_comparison(CF_0_45_4_Dry_AT, 'CF_0_45_4_Dry_AT', CF_0_45_4_Dry_IT, 'CF_0_45_4_Dry_IT')
```



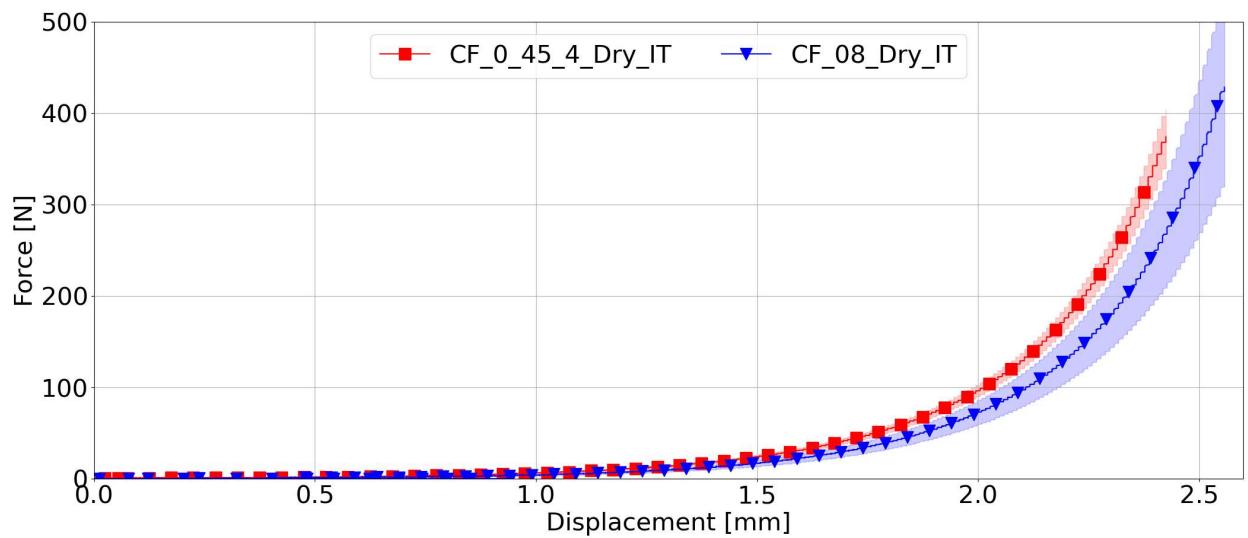
```
In [97]: plot_comparison(CF_0_45_4_Dry_AT, 'CF_0_45_4_Dry_AT', CF_0_45_4_20cst_AT, 'CF_0_45_4_20cst_AT')
```



```
In [93]: plot_comparison(CF_0_45_4_Dry_AT, 'CF_0_45_4_Dry_AT', CF_08_Dry_AT, 'CF_08_Dry_AT')
```



```
In [91]: plot_comparison(CF_0_45_4_Dry_IT, 'CF_0_45_4_Dry_IT', CF_08_Dry_IT, 'CF_08_Dry_IT')
```



```
In [ ]:
```