Experiment validation

April 9, 2025

0.1 Init. data

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
from CONFIG import *
from RobotModel import*
from SoilModel import*
import pandas as pd
%matplotlib inline
```

[ ]:
```
SELECT_DENSITY = DENSITY_1_42

fineSand = SoilModel("fine sand",
                         bulk_density=SAND_FINE_BULK_DENSITY[SELECT_DENSITY],
                         particle_density = SAND_FINE_GRAIN_DENSITY,
                         interface_friction = SAND_FINE_INTERFACE_FRICTION,
                         friction = SAND_FINE_FRICTION,
                         tank_H = TANK_HEIGHT,
                         tank_D = TANK_DIAMETER,
                         e_min = SAND_FINE_VOIDRATIO_MIN,
                         e_max = SAND_FINE_VOIDRATIO_MAX)

coarseSand = SoilModel("coarse sand",
                         bulk_density=SAND_COARSE_BULK_DENSITY[SELECT_DENSITY],
                         particle_density = SAND_COARSE_GRAIN_DENSITY,
                         interface_friction = SAND_COARSE_INTERFACE_FRICTION,
                         friction = SAND_COARSE_FRICTION,
                         tank_H = TANK_HEIGHT,
                         tank_D = TANK_DIAMETER,
                         e_min = SAND_COARSE_VOIDRATIO_MIN,
                         e_max = SAND_COARSE_VOIDRATIO_MAX)
```

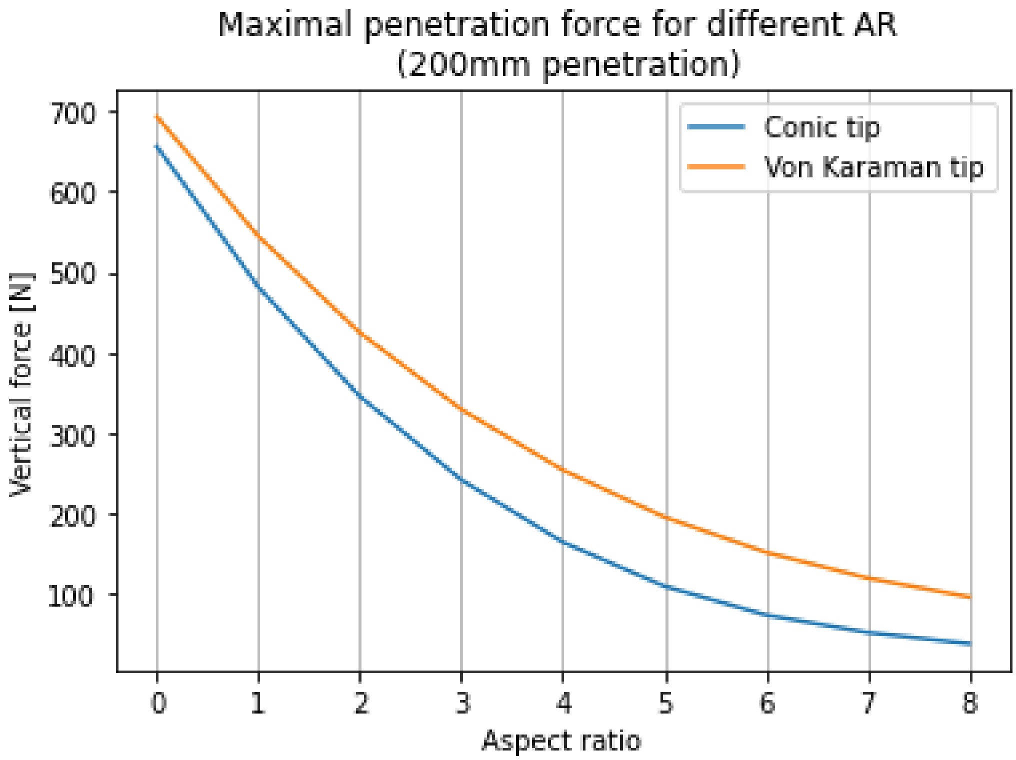
```
M_solid  48.17946493545307
V_solid  0.018968293281674438
V_total  0.03392920065876977
Dr [%] =  50.33358042994812  e =  0.7887323943661971
M_solid  48.17946493545307
V_solid  0.018968293281674438
V_total  0.03392920065876977
```

```
Dr [%] =  3.2193158953722607  e =  0.7887323943661971
```

0.2 Comparison of tip shapes

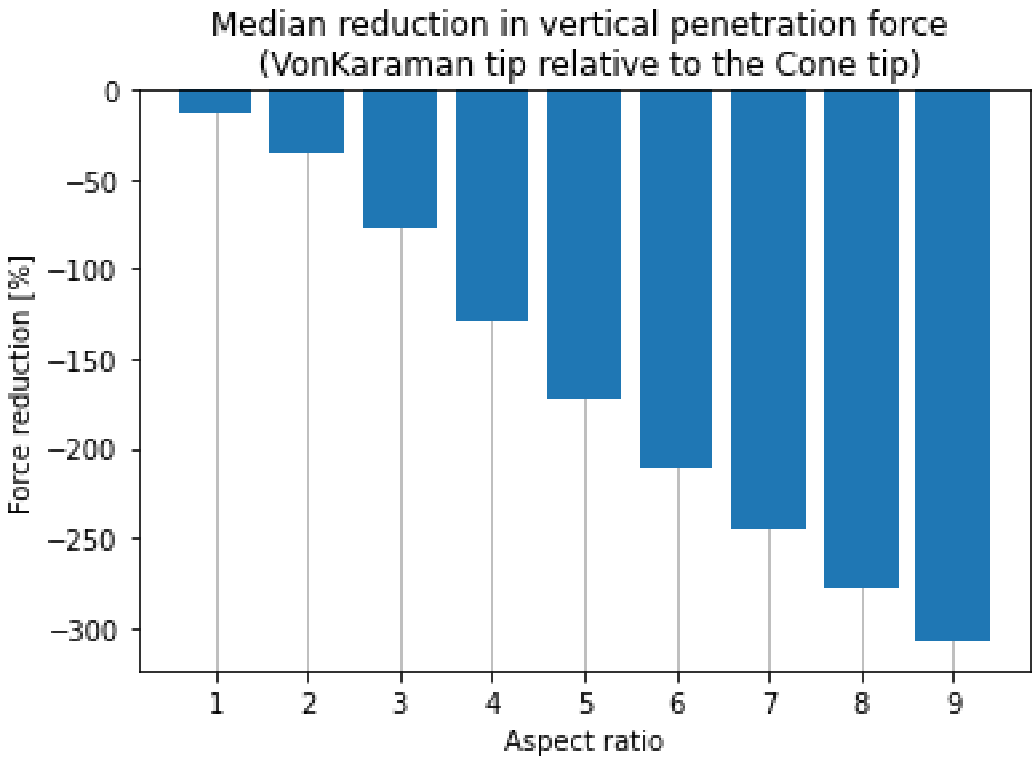
[ ]:
```
# Definition of the tip shapes

aspect_ratios = np.arange(1,10)
conic_res= []
karaman_res = []
mean_diff = []

for ar in aspect_ratios:
    tip_cpt = robotModel(radius=15,AR=ar,profile="Cone", body_length=200)
    tip_karaman = robotModel(radius=15,AR=ar,profile="VonKaraman",␣
 ↪body_length=200)

    sim_cpt = fineSand.getInterraction(tip_cpt,200)
    sim_karaman = fineSand.getInterraction(tip_karaman,200)

    conic_res.append(np.max(sim_cpt['force_z']))
    karaman_res.append(np.max(sim_karaman['force_z']))

    delta = (sim_cpt['force_z'] - sim_karaman['force_z'])/sim_cpt['force_z']
    delta = delta.fillna(0)
    mean_diff.append(np.median(delta))

    # plt.plot((sim_cpt['force_z'] - sim_karaman['force_z'])/sim_cpt['force_z'])
    # plt.show()
    # exit()

conic_res= np.array(conic_res)
karaman_res = np.array(karaman_res)
mean_diff = np.array(mean_diff)
```

[ ]:
```
fig, ax = plt.subplots()
ax.plot(conic_res)
ax.plot(karaman_res)
plt.draw()
ax.set_title('Maximal penetration force for different AR \n (200mm␣
 ↪penetration)')
ax.set_xlabel('Aspect ratio')
ax.set_ylabel('Vertical force [N]')
ax.legend(['Conic tip','Von Karaman tip'])
ax.grid(axis='x')
```



We can see that the maximal force required during the vertical penetraiton is always greater for Von Karaman profile compared to the Conic tip used for CPT probes.

[ ]:
```
fig, ax = plt.subplots()
ax.bar(np.array(aspect_ratios),mean_diff*100,zorder=2)
ax.set_title('Median reduction in vertical penetration force \n (VonKaraman tip␣
 ↪relative to the Cone tip)')
ax.set_xlabel('Aspect ratio')
ax.set_ylabel('Force reduction [%]')
ax.set_xticks(aspect_ratios)
ax.grid(axis='x',zorder=1)
#plt.hist(conic_res);
#plt.hist(karaman_res);
plt.draw()
```



The negative values for "reduction in vertical penetration force" show that, on average, along a 200 mm penetration, the Von Karaman tip requires more force than the conical tip to penetrate to the same depth.

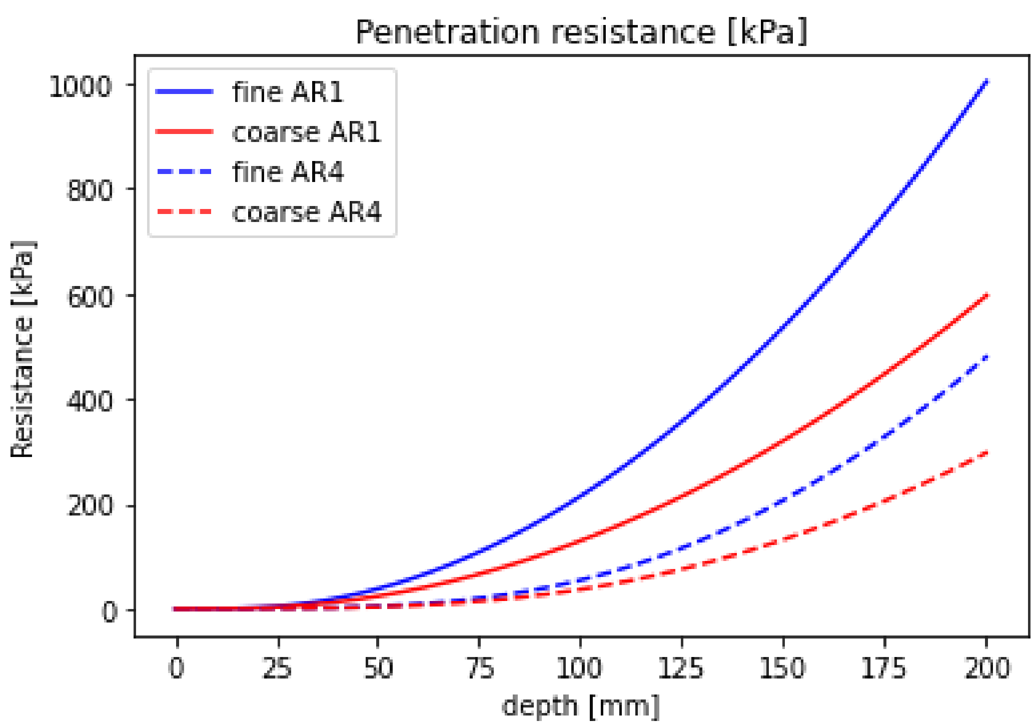
0.3 Comparison of soil properties

[ ]:
```
tip_karaman_1 = robotModel(radius=15,AR=1,profile="VonKaraman", body_length=200)
tip_karaman_4 = robotModel(radius=15,AR=4,profile="VonKaraman", body_length=200)

sim_fineSand_1 = fineSand.getInterraction(tip_karaman_1,200)
sim_coarseSand_1 = coarseSand.getInterraction(tip_karaman_1,200)

sim_fineSand_4 = fineSand.getInterraction(tip_karaman_4,200)
sim_coarseSand_4 = coarseSand.getInterraction(tip_karaman_4,200)
```

[ ]:
```
area = np.pi * (15e-3)**2

x_1_1 = sim_fineSand_1['depth']
y_1_1 = (sim_fineSand_1['force_z'] + sim_fineSand_1['skinFriction']) /area *1e-3
x_2_1 = sim_coarseSand_1['depth']
y_2_1 = (sim_coarseSand_1['force_z'] + sim_coarseSand_1['skinFriction']) /area␣
↪*1e-3
```

```
x_1_4 = sim_fineSand_4['depth']
y_1_4 = (sim_fineSand_4['force_z'] + sim_fineSand_4['skinFriction']) /area *1e-3
x_2_4 = sim_coarseSand_4['depth']
y_2_4 = (sim_coarseSand_4['force_z'] + sim_coarseSand_4['skinFriction']) /area␣
↪*1e-3



fig, ax = plt.subplots()
ax.plot(x_1_1,y_1_1,color='blue');
ax.plot(x_2_1,y_2_1,color='red');

ax.plot(x_1_4,y_1_4,'--',color='blue');
ax.plot(x_2_4,y_2_4,'--',color='red');
ax.set_xlabel("depth [mm]")
ax.set_ylabel("Resistance [kPa]")
ax.set_title("Penetration resistance [kPa]")
ax.legend(['fine AR1','coarse AR1','fine AR4','coarse AR4'])
```

[ ]: <matplotlib.legend.Legend at 0x21b104b11e0>



[ ]:

```

```