Wilhelminakanaal_zettingen

January 3, 2017

T.N.Olsthoorn

1 Wilhelminakanaal

1.1 Wat is kans op zettingsverschillen afhankelijk van de afstand tussen punten?

Bij de te verwachting schade door grondwaterstandsdaling zijn niet de zettingen als zodanig van belang maar juist de zettingsverschillen. De gebruikelijke manier om die in beeld te brengen is door berekening van een semi-variogram waarbij de varantie van het zettingsverschil tussen punten wordt berekend afhankelijk hun tussenafstand. Voor gemakkelijker begrip wordt hieronder echter de standaardafwijking van zettingsverschillen berekend afhankelijk van de afstand tussen de punten. Dit komt op hetzelfde neer.

Berekening verloopt als volgt: + bereken voor alle puntparen (sonderingen) waarvoor een totaalzetting is gegeven hun tussenafstand + sorteer de lijst en verdeel de punten in bundeltjes met van tevoren gekozen tussenafstandsgrenzen. + reken voor elke bundel (dus onderlinge afstand) de standaardafwijking van het zakkingsverschil uit van de punten paren in de bundel. + maak een grafiek van de standaardafwijking van deze zakkingsverschillen tegen de puntenafstand. + schat in, door extrapolatie naar tussenafstand nul, wat de standaardafwijking is nabij nul, zodat een beeld wordt verkregen van de kans op zakkingsveschillen tussen punten binnen een gebouw.

1.2 Inlezen gegevens

Hierons staat de lijst toegestuurde Bestanden. Alleen het eerste wordt gebruikt, want dat bevat de totale zetting. Verschilzettingen worden verderop berekend.

```
In [196]: !ls
```

```
SC001_KaleIngreep_MaaiveldzettingOpSondeerlocaties_v001.csv
SC001_KaleIngreep_ZettingsverschilTussenSondeerlocaties_100m_v001.csv
SC001_KaleIngreep_ZettingsverschilTussenSondeerlocaties_v001.csv
WHK-250-10000-00-MEM-ALG-01.pdf
Wilhelminakanaal_zettingen.ipynb
```

```
In [145]: # Importeer benodigde functionaliteit
          import csv
          import numpy as np
          import matplotlib.pyplot as plt
```

```
In [146]: # Lees het bestand in, en maak er een array van 3 kolommen van [eindzetti
          with open('SC001_KaleIngreep_MaaiveldzettingOpSondeerlocaties_v001.csv')
              rdr = csv.reader(csvfile, delimiter=',', quotechar='"')
              data = np.zeros((10000,3))
              for ir, row in enumerate(rdr):
                  if ir>0:
                      data[ir,:] = np.array([float(row[1]), float(row[2]), float(ro

              # bewaar alleen de rijen met data
              dat=data[1:ir+1]
```

1.3 Uitwerking

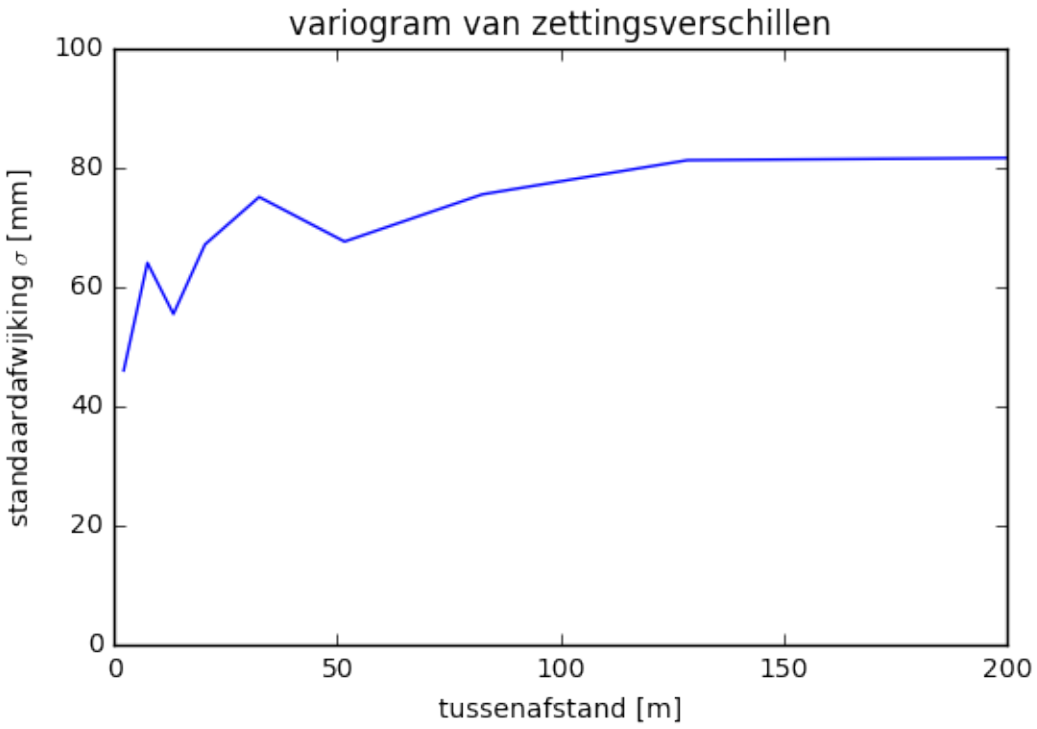
- bereken tussenafstanden
- bereken verschilzettingen
- verwijder dubbelingen want (i,j) = (j,i)

```
In [147]: Np = dat.shape[0] # aantal punten in bestand
          ip = np.arange(Np, dtype=int) # puntnummers, om later te kunnen plotten
          d = dat[:,0]  # eindzetting
          x = dat[:,1]  # x-coordinaat
          y = dat[:,2]  # y-coordinaat

          # combineer, dus kolom wordt nu aan array (Np, Np) van punt combinaties
          Ip1 = ip[np.newaxis,:] - np.zeros((Np, 1), dtype=int)  # puntnummers alle
          Ip2 = ip[:,np.newaxis] - np.zeros((1, Np), dtype=int)  # puntnummers alle
          Dx = x[np.newaxis, :] - x[:,np.newaxis]  # x[i] - y[j] voor alle puntenpa
          Dy = y[np.newaxis, :] - y[:,np.newaxis]  # y[i] - y[j] voor alle puntenpa
          R = np.sqrt( Dx**2 + Dy**2)      # De stand tussen elk puntenpaar
          D = (d[np.newaxis,:] - d[:,np.newaxis]) # Het zettingsverschil tussen elk

          # Zet alle R onder de diagonaal -1 om ze eruit te kunnen gooien want dubb
          for i in range(R.shape[0]):
              R[i, np.arange(i)]=-1
```

- maak van de arrays lijsten
- sorteer op tussenastand
- onthoud de puntnummers om ze later te kunnen tekenen
- beperk de tussenafstanden tot een redelijke waarde van zeg 1500 m
- kies oplopende reeks tussenafstanden om bundeltjes van punten paren te maken
- bereken de standaardafwijking van de zettingsverschillen voor elke bundel
- plot standaardafwijking zettingsverschil tegen tussenafstand van bundel

```
In [197]: # sorteren van paren op onderlinge afstand en selecteren
          I = np.argsort(R.ravel())  # Maak lijst van afstanden en lever sorteer in
          r = R.ravel()[I]           # Geef lijst gesorteerde afstanden
          d = D.ravel()[I]           # Geef lijst van bijbehorende zettingsverschil
          ip1 = Ip1.ravel()[I]      # lijst van punt 1 van puntenpaar
          ip2 = Ip2.ravel()[I]      # lijst van punt 2 van puntenpaar
```

```
# selecteer alleen relevanten afstanden, zeg < 1500 m
J = np.logical_and(r>0, r<1500)
iip1 = ip1[J] # punt 1 van puntenpaar
iip2 = ip2[J] # punt 2 van puntenpaar
dd   = d[  J] # neem alleen afstanden die relevant zijn
rr   = r[  J] # zelfde

# Kies een reeks oplopende afstanden om punten te kunnen bundelen op tuss
h = np.hstack((0, 5, np.logspace(1, 3, 11)))    # kies tussen afstanden
hm = 0.5 * (h[:-1] + h[1:])  # gemidelde van de tussenafstanden van elke

# Bereken standaardafwijking zettingsverschil van elke bundel per tussena
sb = np.zeros(len(h)-1,)  # initialiseer de standaardafijkingen op nul
hb = np.zeros(len(h)-1,)  # initialiseer de standaardafijkingen op nul
I1 = list()
I2 = list()
for ib in range(len(hm)):  # loop alle bundeltjes af
    J = np.logical_and(rr>=h[ib], rr<h[ib+1]) # indices punten binnen bun
    sb[ib] = np.std(dd[J]) # bereken stdafwijking van bundel
    hb[ib] = np.mean(rr[J]) # gemiddelde afstand puntenparen in bundel
      # dit is beter dan het gemiddelde van de onder en bovengrens van de
    I1.append(iip1[J]) # onthoud eeste punt van puntenparen in bundel
    I2.append(iip2[J]) # onthoud tweede punt van puntenparen in bundel

# Presenteer de resltaten
ax = plt.figure().add_subplot(111)
ax.set(xlabel='tussenafstand [m]', ylabel=r'standaardafwijking $\sigma$ [
       ylim=(0, 100), title='variogram van zettingsverschillen')
ax.plot(hb, sb) # plot resultaten
plt.show() # laat plot zien
```



- Controleer het resultaat door voor de eerste puntenparen van de lijst de tussenafstand en het zettingsverschil te laten zien.

```
In [171]: N = 50  # bijv.  voor de eerste 50 punten.
          np.vstack((rr[:N], dd[:N])).T

Out[171]: array([[   1.        ,   67.        ],
                 [   1.        ,    0.        ],
                 [   1.        , -119.        ],
                 [   1.        ,   -1.        ],
                 [   1.        ,    0.        ],
                 [   1.        ,    0.        ],
                 [   1.41421356,    1.        ],
                 [   1.41421356,    0.        ],
                 [   1.41421356,  113.        ],
                 [   2.23606798,    0.        ],
                 [   3.16227766,    1.        ],
                 [   4.12310563,   -1.        ],
                 [   4.12310563,   -1.        ],
                 [   4.47213595,    0.        ],
                 [   4.47213595,   24.        ],
                 [   5.        ,  -42.        ],
                 [   5.09901951,  -10.        ],
                 [   5.65685425,   -1.        ],
```

```
[   6.32455532,  -61.        ],
[   6.40312424,  -76.        ],
[   6.70820393,  -14.        ],
[   7.21110255,   79.        ],
[   8.48528137,    0.        ],
[   8.94427191,    0.        ],
[   8.94427191, -205.        ],
[   9.05538514,    0.        ],
[   9.8488578 ,  -78.        ],
[   9.8488578 ,    0.        ],
[  10.04987562,    0.        ],
[  10.29563014,    1.        ],
[  10.44030651,   -1.        ],
[  10.77032961,    0.        ],
[  10.77032961,    0.        ],
[  10.77032961,   36.        ],
[  10.81665383,    0.        ],
[  11.        ,   27.        ],
[  11.04536102, -105.        ],
[  11.40175425,  -85.        ],
[  11.40175425,    1.        ],
[  12.        , -172.        ],
[  12.08304597,   -1.        ],
[  12.20655562,  -56.        ],
[  12.36931688,  150.        ],
[  12.72792206, -149.        ],
[  12.72792206,   -2.        ],
[  12.80624847,    6.        ],
[  13.        ,    6.        ],
[  13.        ,    0.        ],
[  13.03840481,  -56.        ],
[  13.34166406,    0.        ]])
```

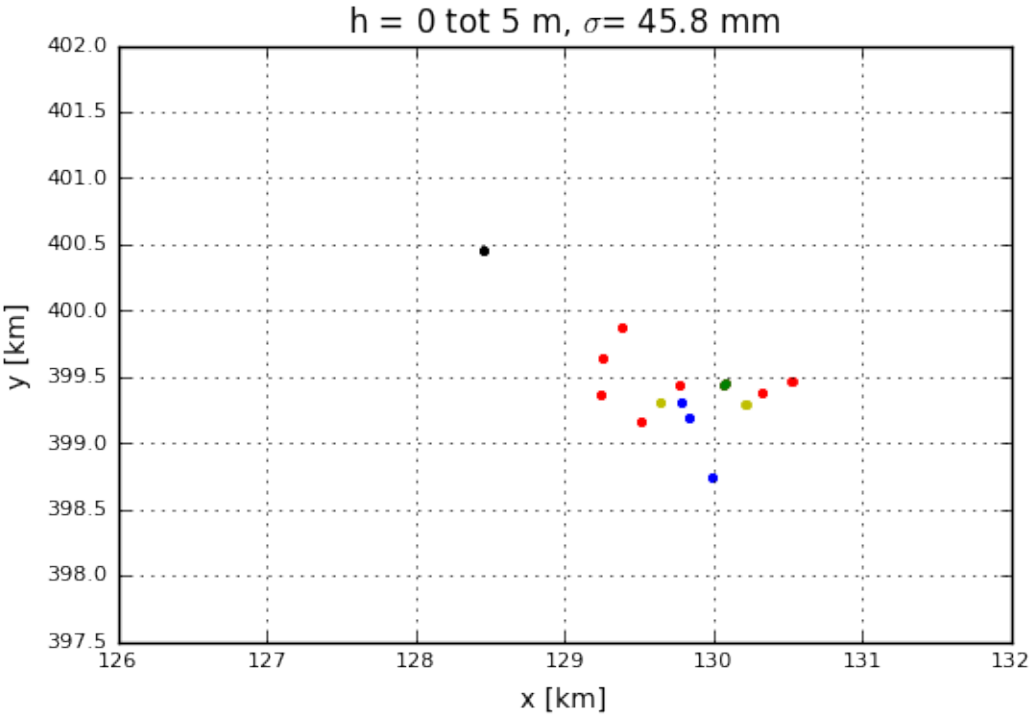
1.4 Resultaat

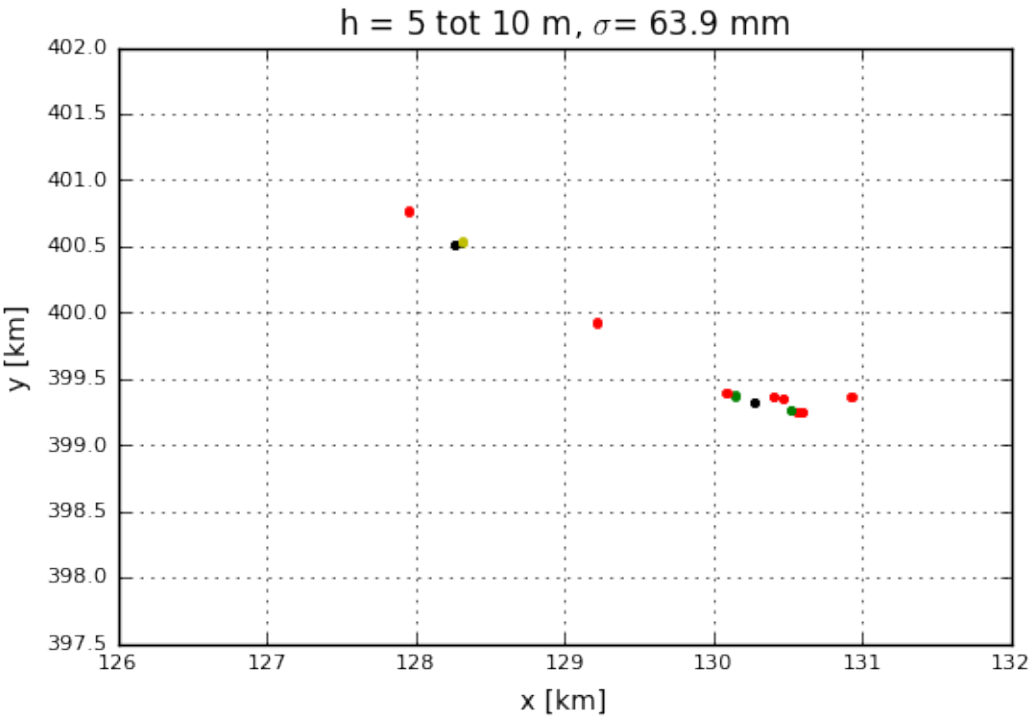
Hieronder wordt het resultaat geverifeerd door per bundel de puntenparen weer te geven en in de titel hun standaardafwijking en de afstanden waartussen de getekende puntenparen vallen.

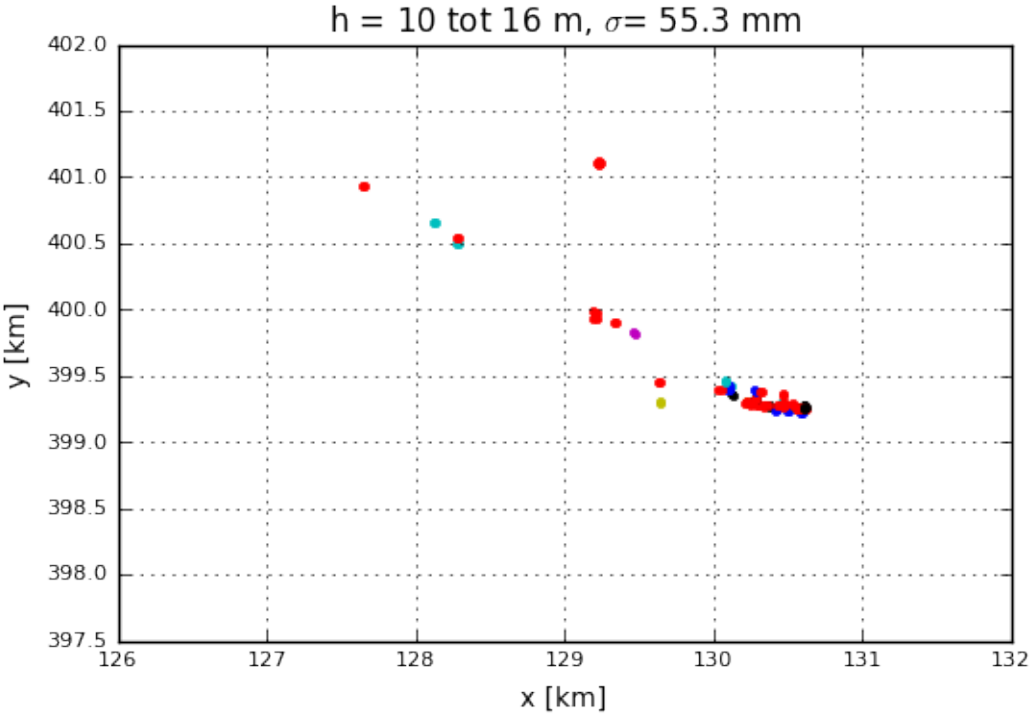
De kleur geeft het zettingsverschil per individueel puntenpaar aan waarbij: + rood = 0-10 mm + blauw = 10-20 mm + groen = 20-30 mm + zwart = 30-40 mm + paars = 40-50 mm + cyaan = 60-70 mm + geel = 70-80 mm

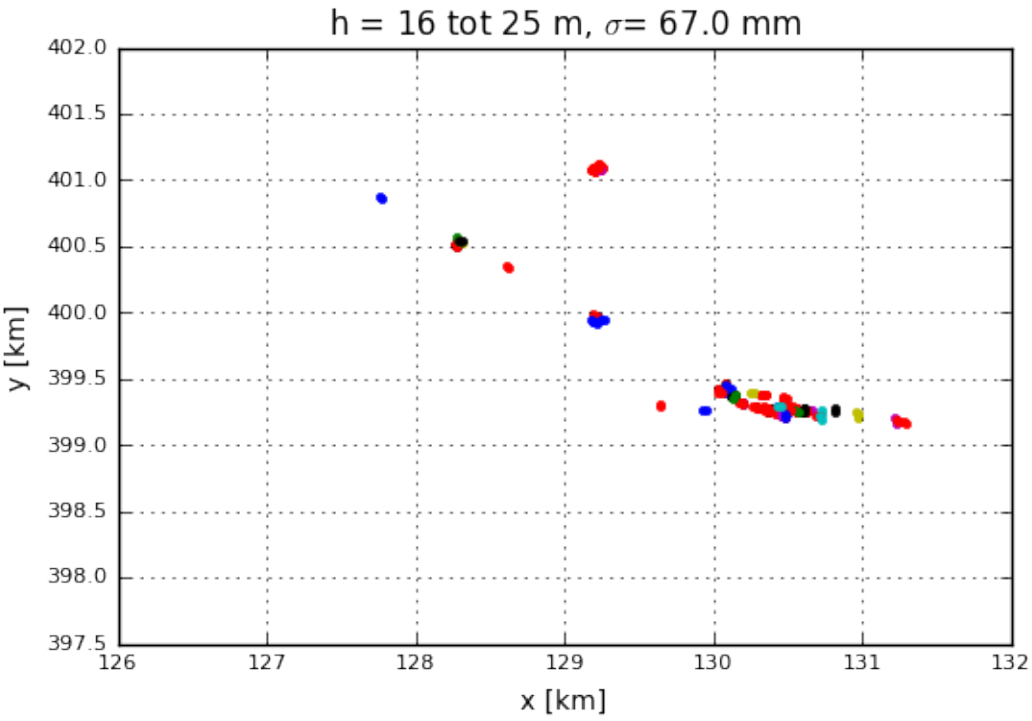
```
In [203]: # Verifeer door de punten paren binnen dezelfde bundel te tekenen
          clrs = 'rbgkmcy'  # kleuren
          Nc = len(clrs)  # aantal kleuren voordat kleuren worden herhaald

          ax = list()  # figurenlijst
          for ib in range(len(sb)):
              ax.append(plt.figure().add_subplot(111)) # nieuwe figuur
```

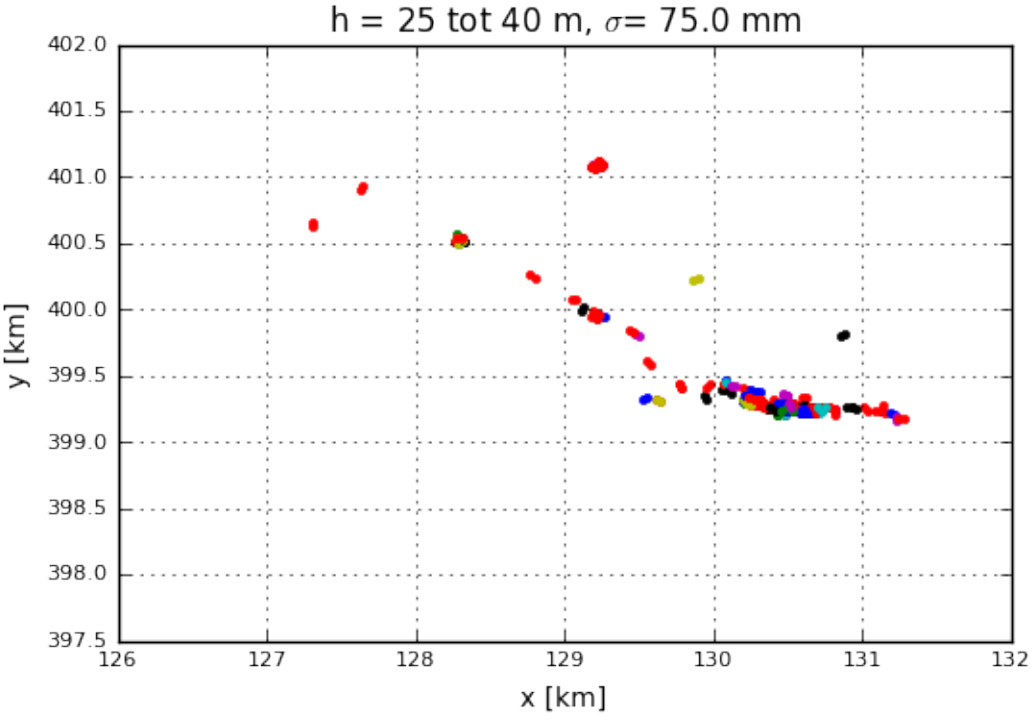
```
        ax[-1].set(xlim=(126, 132), ylim=(397.5, 402), xlabel='x [km]', ylabe
        plt.setp(ax[-1].get_xticklabels(), fontsize=8)
        plt.setp(ax[-1].get_yticklabels(), fontsize=8)
        ax[-1].set_title(r'h = {:.0f} tot {:.0f} m, $\sigma$= {:.1f} mm'.form
        ax[-1].grid(True)

        for j, k in zip(I1[ib], I2[ib]):
            ax[-1].plot(x[[j, k]]/1000., y[[j, k]]/1000., '.-', color=clrs[ir
plt.show()
```

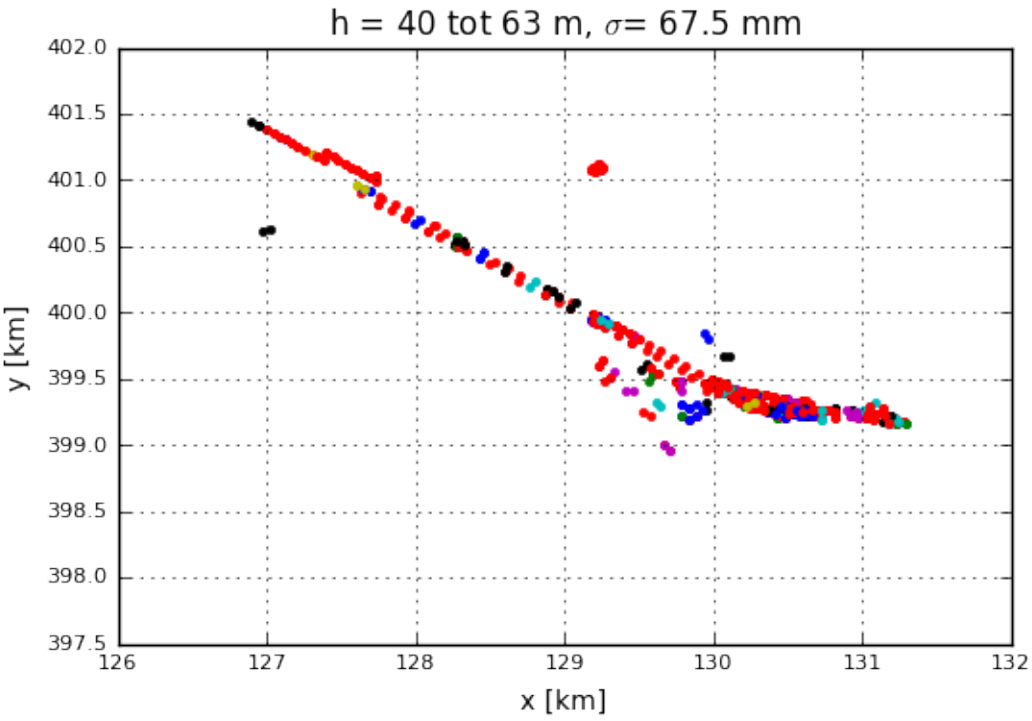


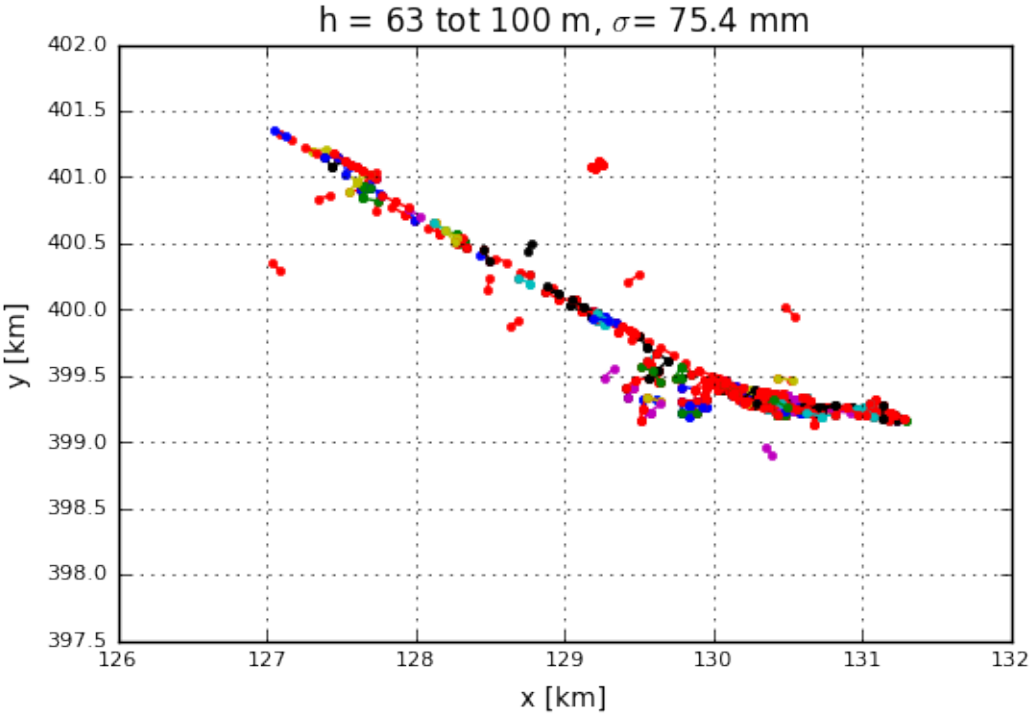

h = 5 tot 10 m, σ= 63.9 mm
y [km]
x [km]
402.0
401.5
401.0
400.5
400.0
399.5
399.0
398.5
398.0
397.5
126
127
128
129
130
131
132

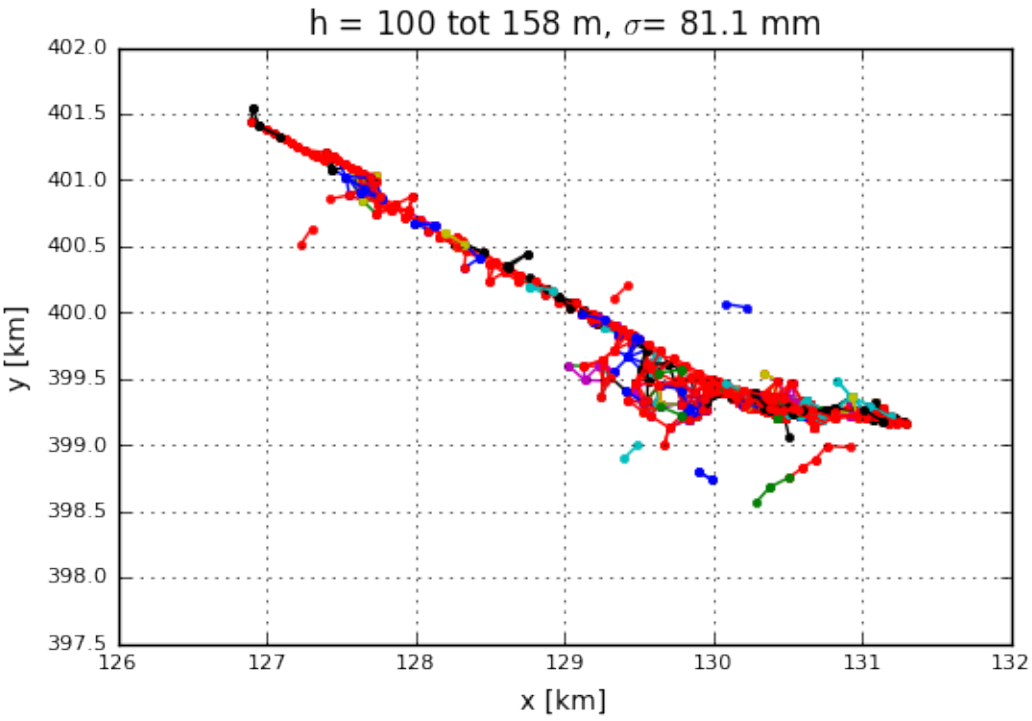

h = 10 tot 16 m, σ= 55.3 mm
y [km]
x [km]
402.0
401.5
401.0
400.5
400.0
399.5
399.0
398.5
398.0
397.5
126
127
128
129
130
131
132

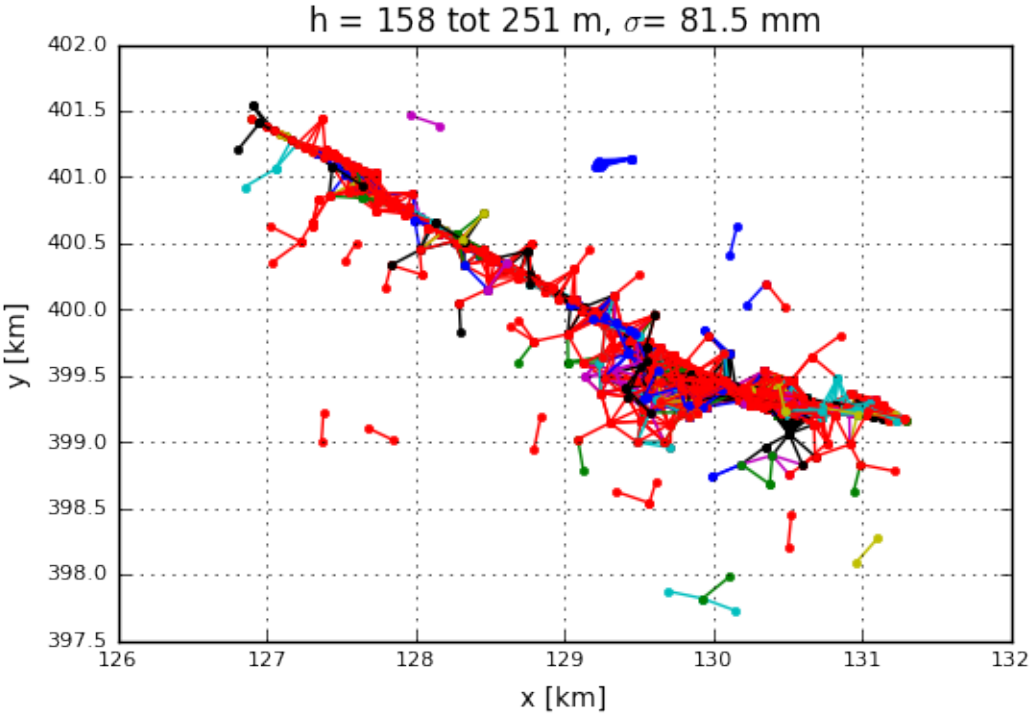

h = 16 tot 25 m, σ= 67.0 mm
y [km]
x [km]
402.0
401.5
401.0
400.5
400.0
399.5
399.0
398.5
398.0
397.5
126
127
128
129
130
131
132

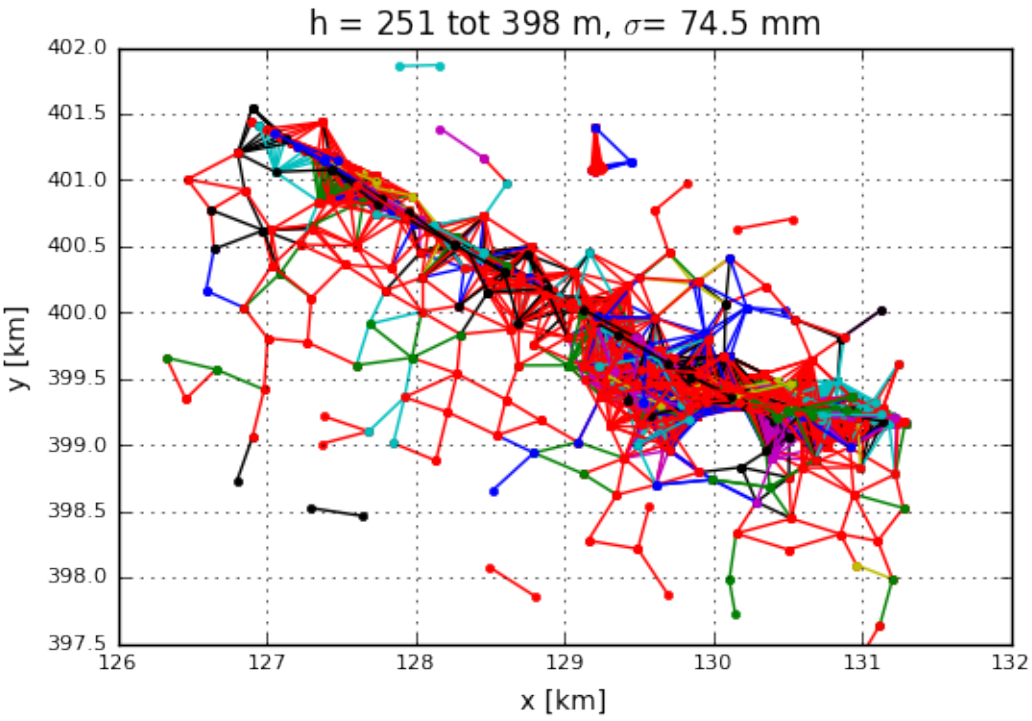

h = 25 tot 40 m, σ= 75.0 mm
y [km]
x [km]
402.0
401.5
401.0
400.5
400.0
399.5
399.0
398.5
398.0
397.5
126
127
128
129
130
131
132

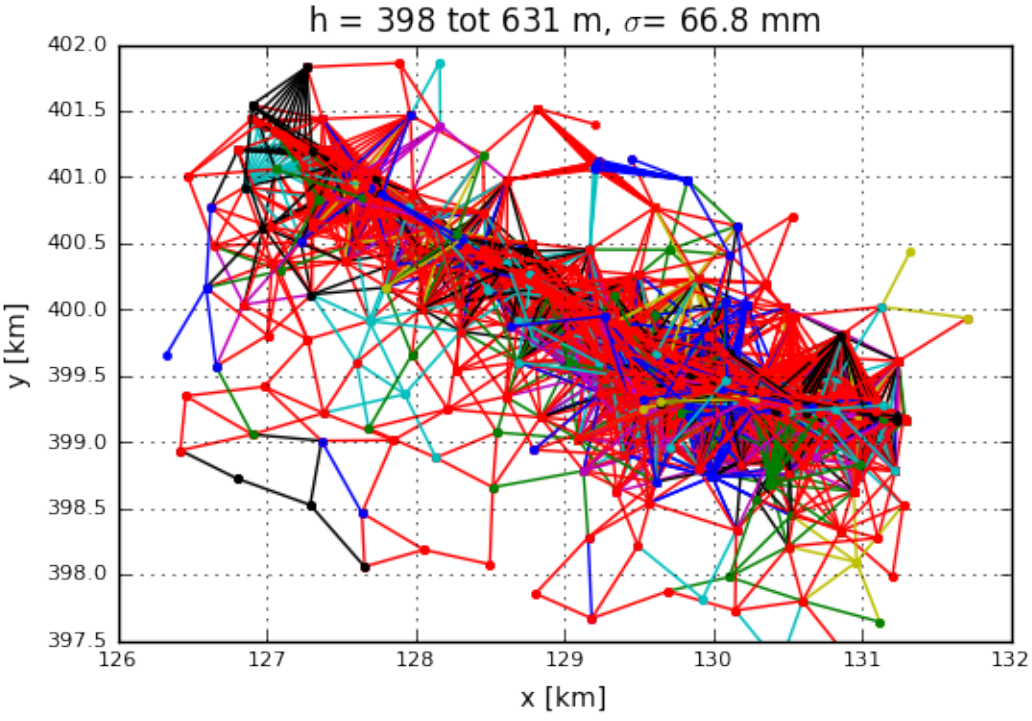

h = 40 tot 63 m, σ= 67.5 mm
y [km]
x [km]
402.0
401.5
401.0
400.5
400.0
399.5
399.0
398.5
398.0
397.5
126
127
128
129
130
131
132

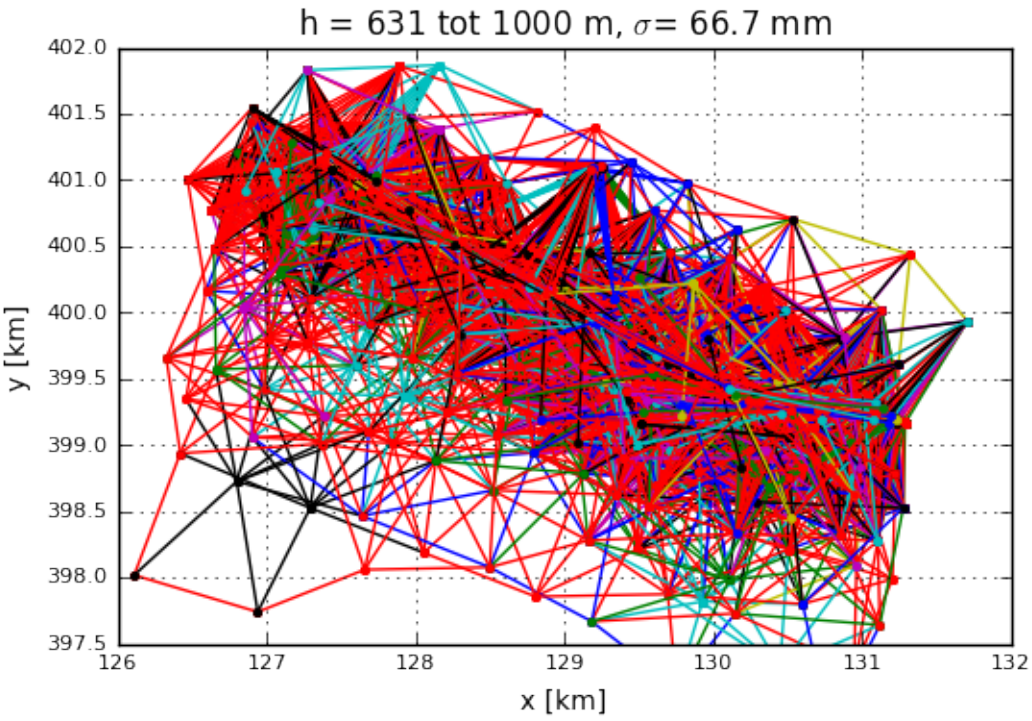

h = 63 tot 100 m, σ= 75.4 mm
y [km]
x [km]
402.0
401.5
401.0
400.5
400.0
399.5
399.0
398.5
398.0
397.5
126
127
128
129
130
131
132


h = 100 tot 158 m, σ= 81.1 mm
y [km]
x [km]
402.0
401.5
401.0
400.5
400.0
399.5
399.0
398.5
398.0
397.5
126
127
128
129
130
131
132


h = 158 tot 251 m, σ= 81.5 mm
y [km]
x [km]
402.0
401.5
401.0
400.5
400.0
399.5
399.0
398.5
398.0
397.5
126
127
128
129
130
131
132


h = 251 tot 398 m, σ= 74.5 mm
y [km]
x [km]
402.0
401.5
401.0
400.5
400.0
399.5
399.0
398.5
398.0
397.5
126
127
128
129
130
131
132


h = 398 tot 631 m, σ= 66.8 mm
y [km]
x [km]
402.0
401.5
401.0
400.5
400.0
399.5
399.0
398.5
398.0
397.5
126
127
128
129
130
131
132



1.5 Conclusie

Op basis van de gegeven zettingen kan niet worden gezegd dat de kans op zettingsverschillen wenzelijk afneemt bij kleinere tussenafstand, de zettingsverschillen nemen immers nauwelijks af met afnemende tussenafsand. Echter, punten met kleine tussenafstanden, dat wil zeggen op een schaal kleiner dan de platte grond van een woning zijn er alleen pal langs het Wilhelminakanaal. Extrapolatie naar naburige (woon)wijk(end) mag alleen als de verwachte geologie daar hetzelfde karakter heeft als die langs het kanaal. Aangezien de bundel punten met korte tussenafstand wel over een grote lengte (van het kanaal) voorkomt, ligt het in de rede dat ook op beperkte afstand van het kanaal hetzelfde beeld aanwezig zal zijn In dat geval kunnen dus ook binnen de (woon)wijk(en) aanzienlijke verschilzettingen optreden. Uitsluiten van verschilzettingen kan dan ook alleen op basis van de informatie onder individuele woningen en niet a priori per straat, wijk of regio.