



Automatisering van modelopbouw met de Tygron watermodule

Emma Kok
11-11-2025

Waarom Waternet en Tygtron?



- Eén integraal beeld van het hele watersysteem
- Snelle berekeningen van wateroverlast en grondwater
- 2D maaiveld ingebouwd
- Ideaal voor gebiedsplannen en stresstesten

Waar staan we nu?

2021
Aanschaf Tygron-
licentie



2022
Waterbalans en
Tygronmodule



2023
Eén werkend
watermodel



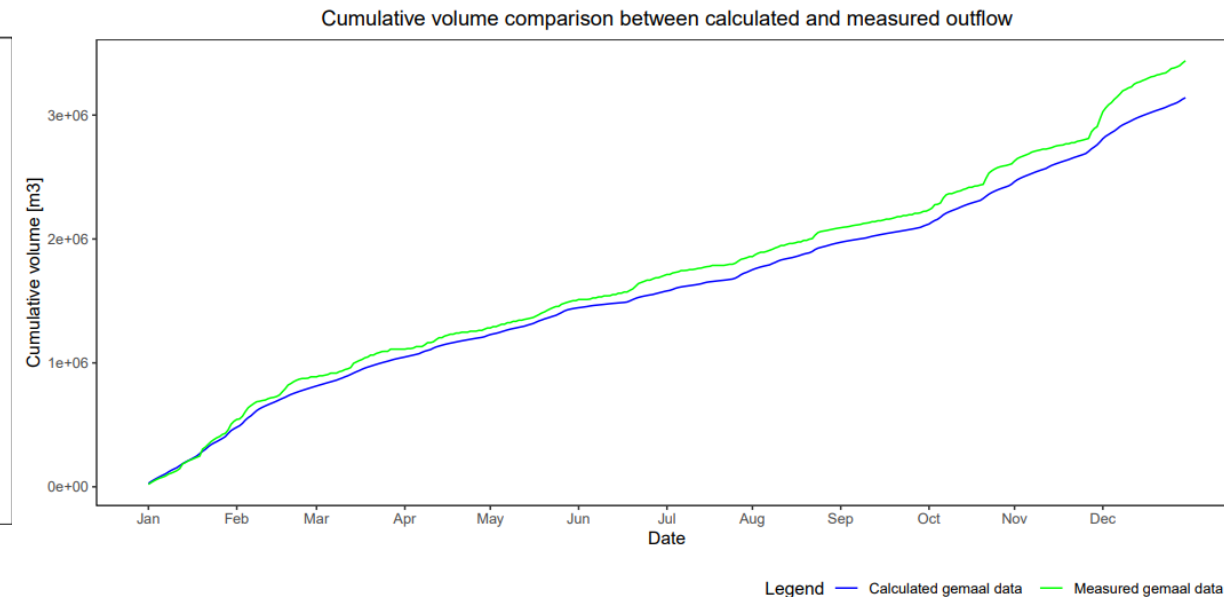
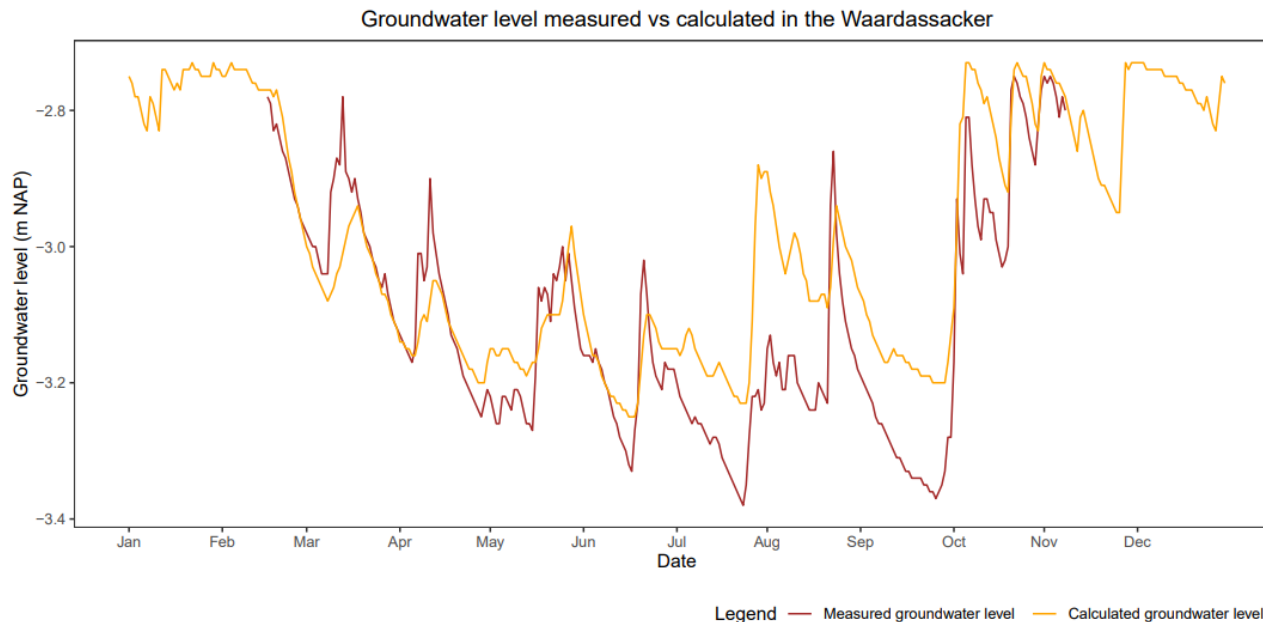
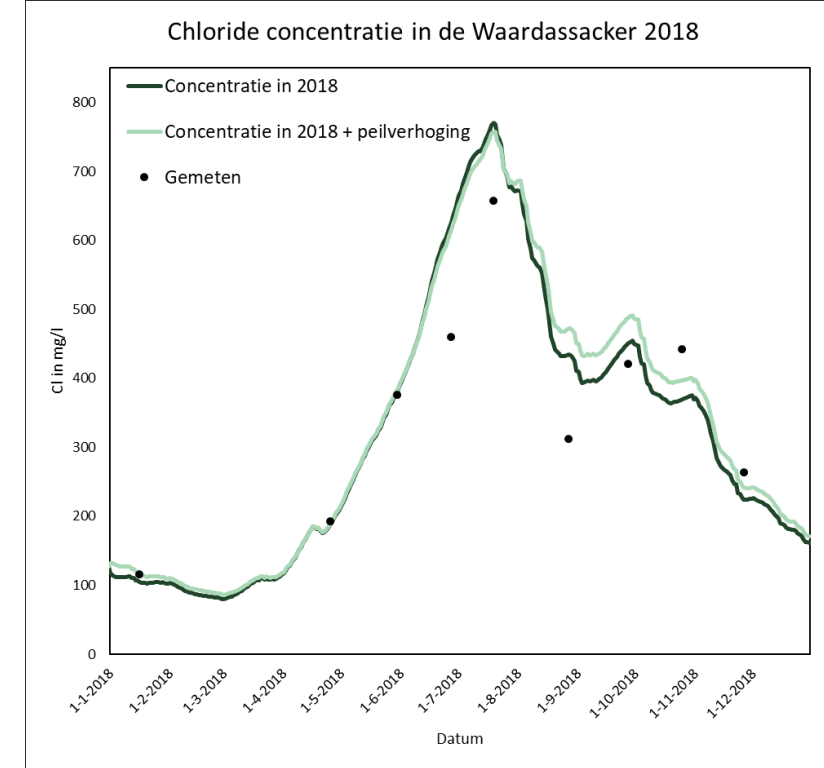
2025
Op kleine schaal
werkende modellen



2026
Werkend watermodel
van een complexe
polder

2023: werkend model

- Veeeeel arbeidsuren + hulp van Tygron
- Daarna nooit meer zo goed gelukt op grote schaal



Paniek

Waterpeilen staan overal op -2,91m! Peilvakken verbonden? > op te lossen door dammen toe te voegen?

Resultaten kloppen nog steeds niet, gemaal lijkt echt niet goed te werken. Misschien ergens nog een bug? Lower en upper treshold verwisselen werkt ook niet.

Inlet capacity attribute verwijderd. Helpt ook niet.

Gemaal Rondehoep als inlaat gemodelleerd. Maakt het niet beter > water lijkt niet uit model te komen.

To do:

- Vierkante duikers toevoegen?

Sifons bleken niet in model te staan. Toegevoegd, maar is dit goed gegaan?



Aantal errors dat duikers geen diameter hebben

Resultaten kloppen nog steeds niet, gemaal lijkt echt niet goed te werken. Misschien ergens nog een bug? Lower en upper treshold verwisselen werkt ook niet.



Script

- Data van HyDamo naar Tygron format
- Aannames op een rijtje
- Duidelijke mappenstructuur

1. Aannames en in- en uitvoerbestanden inlezen

Aannames

```
# aanname stuwbreedte wanneer er geen breedte bekend is
stuwbreedte_aanname = 1 # in meter
inlaat_Q = 0.0016 # in m3/s
duikerdiameter_aanname = 0.8 # in meter
weir_coeff = 0.865
hoogte_onder_mv_afwijkende_pg = 0.30 # aangenomen drooglegging in afwijkende peilgebieden
winterpeil_boezem = -0.40
zomerpeil_boezem = -0.40
aanname_damhoogte = 0.50 # aantal meter dat keerwanden boven streefpeil liggen
aanname_ruwheid_duiker = 0.014 # default tygron

afwijkend_peil_aanwezig = True # zet op True als je afwijkende peilgebieden hebt. Anders zet op False
peil = 'Winter_peil' # Als je zomer peil wil, dan 'Zomer_peil' invullen
```

In- en Uitvoerbestanden

```
# Invoerbestanden
invoer_map = r"C:\Users\Kok30\OneDrive - Waternet Amsterdam\Documents\Projecten\01 Landelijke Hydrologie\09 Tygron\02Atek_polder\01_GIS\01_shapefiles\invoer_script"
uitvoer_map = r"C:\Users\Kok30\OneDrive - Waternet Amsterdam\Documents\Projecten\01 Landelijke Hydrologie\09 Tygron\02Atek_polder\01_GIS\01_shapefiles\uitvoer_script"
mapnaam_uitvoer = datetime.today().strftime('%Y-%m-%d') # submap aanmaken zodat je niet meteen oude dingen overschrijft

# Invoerpaden
# peilvakken, let op: voorbewerkingen nodig in QGIS! (repair geometries, zie hoofdstuk 2)
bestand_peilvakken = invoer_map + '/peilvakken_boezem.shp'
pad_stuwen = invoer_map + '/stuwen_punt.shp'
pad_keerwanden_pnt = invoer_map + '/keerwanden_punt.shp'
pad_keerwanden_ln = invoer_map + '/keerwanden.shp'
pad_duikers = invoer_map + '/duikers.shp'
pad_sifons = invoer_map + '/sifons.shp'
pad_afwijkende_peilgebieden = invoer_map + '/mvhoogte_afwijkpeil.shp'
pad_meetpunten = invoer_map + '/meetpunten.shp'
```

Script

- Omschrijven van kolommen
- 'Simpele' code
- Gemalen zelf intekenen

```
# Code en geometry apart bewaren, zodat die data behouden blijft
# Alleen gewone duikers selecteren, de andere duikers worden nu al meegenomen in het bestand met de inlaten
duikers = duikers[duikers.KWKINLAT == 2] #alleen gewone duikers meenemen

#selectie maken van de data die belangrijk is voor ons
duikers_geom = duikers[['KDUIDENT', 'KDUVORM', 'KDUHGA1', 'KDULENGT', 'KDUBREED', 'KDUBOKBO', 'KDUBOKBE', 'geometry' ]]

#Omdat er 'gewone duikers' zijn die ook peilvakgrenzen kruisen, moeten we deze duikers een speciale waarde geven voor de hoogte
#Gewoonlijk zijn deze duikers haakse bochten, en kan het water via deze duikers stromen zodra het waterpeil in het hogere peilvak het streefpeil bereikt
#Daarom controleren we welke duikers peilvakgrenzen kruisen met een buffer van 1 meter
duikers['geometry'] = duikers.buffer(1, cap_style=3)
duikers_peilvakken = duikers.sjoin(peilvakken,how='left', predicate='overlaps')

#Echter, we moeten ook de waterstanden BO van alle duikers definiëren om de KDUBOKBO te bepalen voor het geval deze niet beschikbaar is
duikers_peilvakken2 = duikers.sjoin(peilvakken,how='left')

#Voor peilscheidende duikers willen we dat de CULVERT_THRESHOLD gelijk is aan het waterpeil van het bovenstroomse peilvak
#De kolom streefpeil geeft alleen informatie over de waterstand voor de duikers die een peilvakgrens kruisen
df_duikers = duikers_peilvakken.groupby('KDUIDENT')[peil].max().reset_index()
df_duikers['streefpeil'] = df_duikers[peil]

#De kolom streefpeil2 geeft informatie over de waterstand voor alle duikers
df_duikers2 = duikers_peilvakken2.groupby('KDUIDENT')[peil].max().reset_index()
df_duikers2['streefpeil2'] = df_duikers2[peil]

#Duikers gegevens aanpassen
duikers_geometry = pd.merge(duikers_geom, df_duikers[['KDUIDENT', 'streefpeil']], on='KDUIDENT')
duikers_geometry = pd.merge(duikers_geometry, df_duikers2[['KDUIDENT', 'streefpeil2']], on='KDUIDENT')

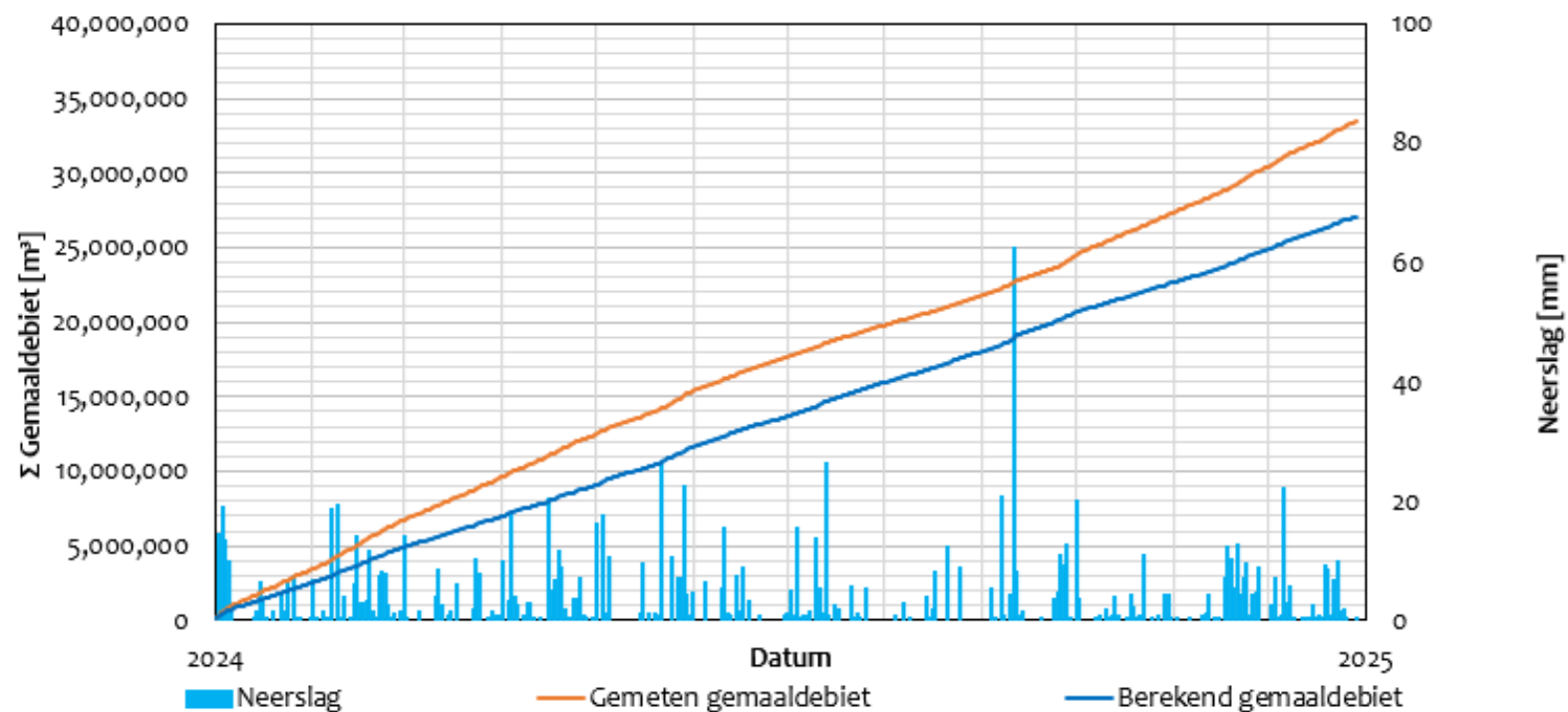
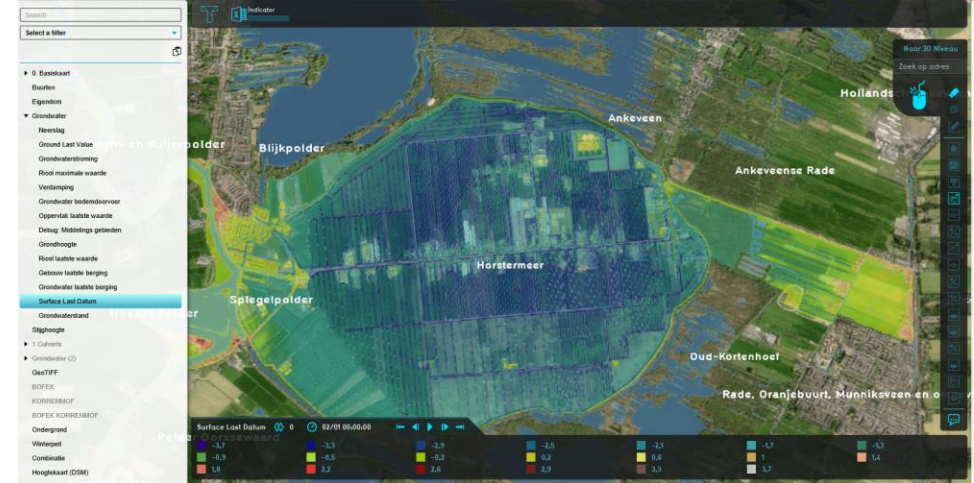
#Als KDUBREED 0 is, maar KDUHGA1 niet 0, dan geldt KDUBREED = KDUHGA1 en vice versa
#Als beide waarden 0 zijn, dan geldt KDUBREED = KDUHGA1 = duikerdiameter_aanname
duikers_geometry.loc[(duikers_geometry['KDUBREED'] == 0) & (duikers_geometry['KDUHGA1'] != 0), 'KDUBREED'] = duikers_geometry['KDUHGA1']
duikers_geometry.loc[(duikers_geometry['KDUHGA1'] == 0) & (duikers_geometry['KDUBREED'] != 0), 'KDUHGA1'] = duikers_geometry['KDUBREED']
duikers_geometry.loc[duikers_geometry['KDUBREED'] == 0, 'KDUBREED'] = duikerdiameter_aanname # protocol hydraulische toets
duikers_geometry.loc[duikers_geometry['KDUHGA1'] == 0, 'KDUHGA1'] = duikerdiameter_aanname # protocol hydraulische toets

#Als de duiker een peilvakgrens kruist, dan is KDUBOKBO=KDUBOKBE=waterpeil in het hogere peilvak
#Vervang KDUBOKBO en KDUBOKBE door de waarden van streefpeil als streefpeil niet NaN is
duikers_geometry['KDUBOKBO'] = duikers_geometry['streefpeil'].where(duikers_geometry['streefpeil'].notna(), duikers_geometry['KDUBOKBO'])
duikers_geometry['KDUBOKBE'] = duikers_geometry['streefpeil'].where(duikers_geometry['streefpeil'].notna(), duikers_geometry['KDUBOKBE'])
```



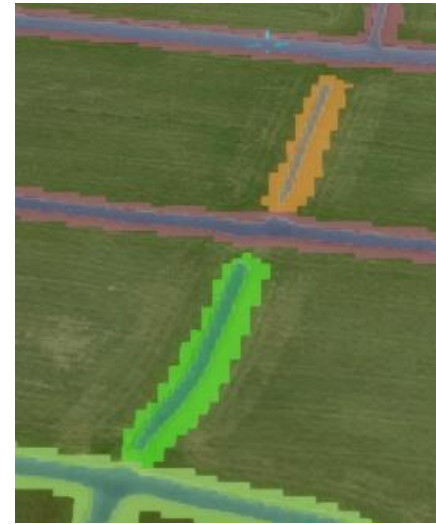
Problemen in een grote polder

- Waar gaat het precies mis?
- Verschillende vervolgstappen



Geïsoleerde watergangen

- Losliggende sloten
- Tygron: via combo overlays multipolygons exporteren waar Tygron denkt dat duikers liggen
- Vervolgens samenvoegen met bestaande duikers



```
aangemaakte_duikers = aangemaakte_duikers.to_crs("EPSG:4326")
peilvakken = peilvakken.to_crs("EPSG:4326")

duikers = aangemaakte_duikers[['geometry']].copy()
duikers['KDUIDENT'] = ['KDU' + str(i + 1) for i in range(len(duikers))]
duikers['KDUKOKBO'] = 0
duikers['KDUKREED'] = 0
duikers['KDUVORM'] = 0

duikers_peilvakken = duikers.sjoin(peilvakken, how='left').copy()
df_duikers = duikers_peilvakken
df_duikers['streefpeil'] = df_duikers['Winter_peil']

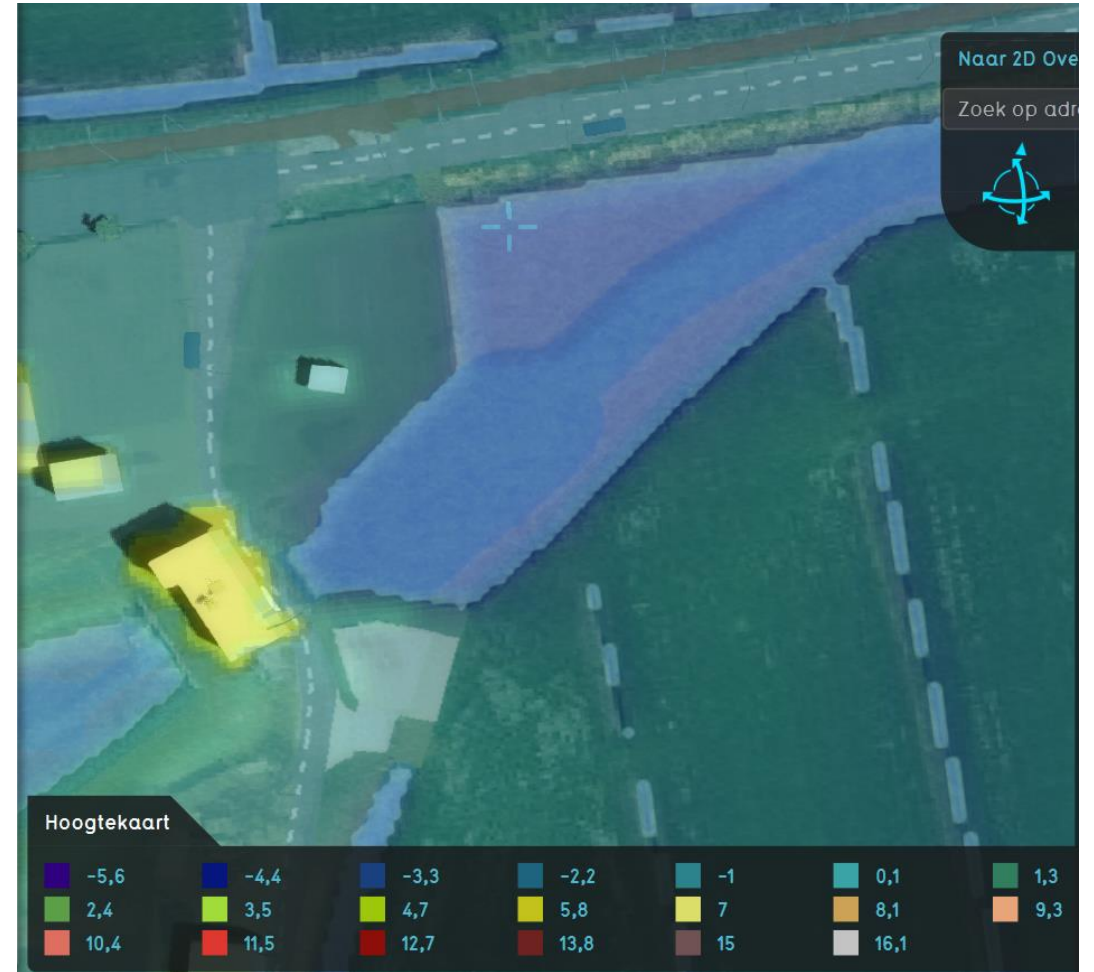
df_duikers['KDUKREED'] = df_duikers['KDUKREED'].astype(float)
df_duikers.loc[df_duikers['KDUKREED'] == 0, 'KDUKREED'] = float(duikerdiameter_aanname)
df_duikers['KDUKOKBO'] = np.where(df_duikers['KDUKOKBO'] == 0, (df_duikers['streefpeil'] - 2/3 * df_duikers['KDUKREED']), df_duikers['KDUKOKBO'])
df_duikers.loc[df_duikers['KDUVORM'] == 99, 'KDUVORM'] = 1
df_duikers.loc[df_duikers['KDUVORM'] == 1, 'KDUVORM'] = 0
df_duikers['ruwheidswa'] = aanname_ruwheid_duiker

df_duikers = df_duikers.rename(columns={
    'KDUKREED': 'CULVERT_DIAMETER',
    'KDUVORM': 'CULVERT_RECT',
    'KDUKOKBO': 'CULVERT_THRES',
    'ruwheidswa': 'CULVERT_N'
})

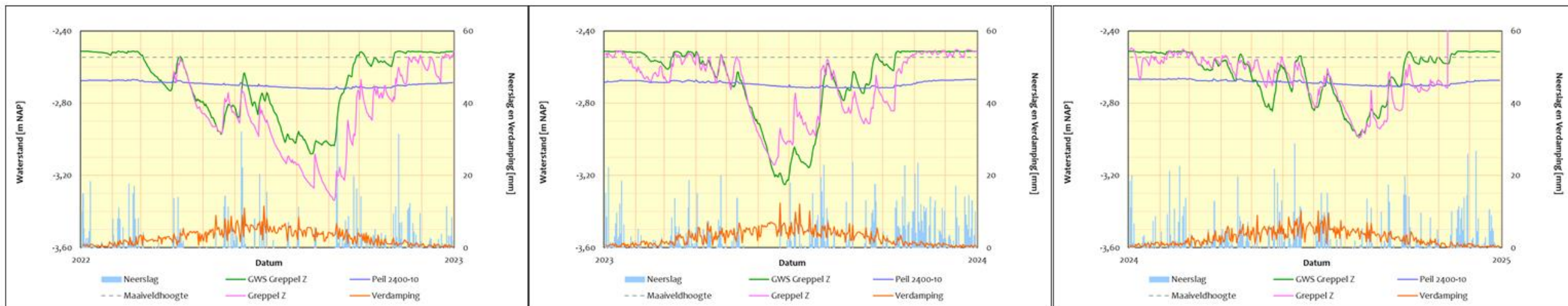
final_df = gpd.GeoDataFrame(df_duikers, geometry= df_duikers['geometry'])
final_df = final_df[['geometry', 'KDUIDENT', 'CULVERT_THRES', 'CULVERT_RECT', 'Winter_peil', 'GPGIDENT', 'streefpeil', 'CULVERT_N', 'CULVERT_DIAMETER']]
final_df = final_df.to_crs("EPSG:4326")
output_duikers = r"C:\Users\Kok30\Downloads\bewerkte_duikers.geojson"
final_df.to_file(output_duikers, driver='GeoJSON')
```

Watergangen met leggerprofielen verbeteren

- Handmatige stappen in GIS
- Volume water kloppend krijgen (fractiesommen)
- Tygron aanname voor secundaire watergangen
- Aangenomen diepte is groter dan de leggerhoogte
 - Rug in het water



Kleine schaal werkend - grondwaterstanden



Groen zijn berekeningen, roze metingen

Einddoel: werkend model

- Van inputdata tot gekalibreerde gevalideerd model
- Nog nodig/wordt aan gewerkt
 - *Kalibratiescripts?*
 - *Waterbalans tool?*



Handleiding opzet watermodel Tygron

Henk van Hardeveld
Emma Kok
Rinke de Groot
Rudolf Koster
Stefan Fritz
Erica Caverzam Barbosa
Joris Westenend

Datum
26 mei 2025

Versie
2025.05

Projectnummers
01.3900 O&I
01.2481 Bodemdaling