



Geautomatiseerd keringen doorrekenen en data verwerken

Joris Westenend

Aanleiding

Levering van overstromingssimulaties in het kader van de ROR3 (Richtlijn Overstromingsrisico's)

Nieuwe normering van regionale keringen

Updaten van de eigen overstromingsbibliotheek

- calamiteiten
- planvorming
- risicokaart eigen assets (drinkwater, riolering, watersysteem)

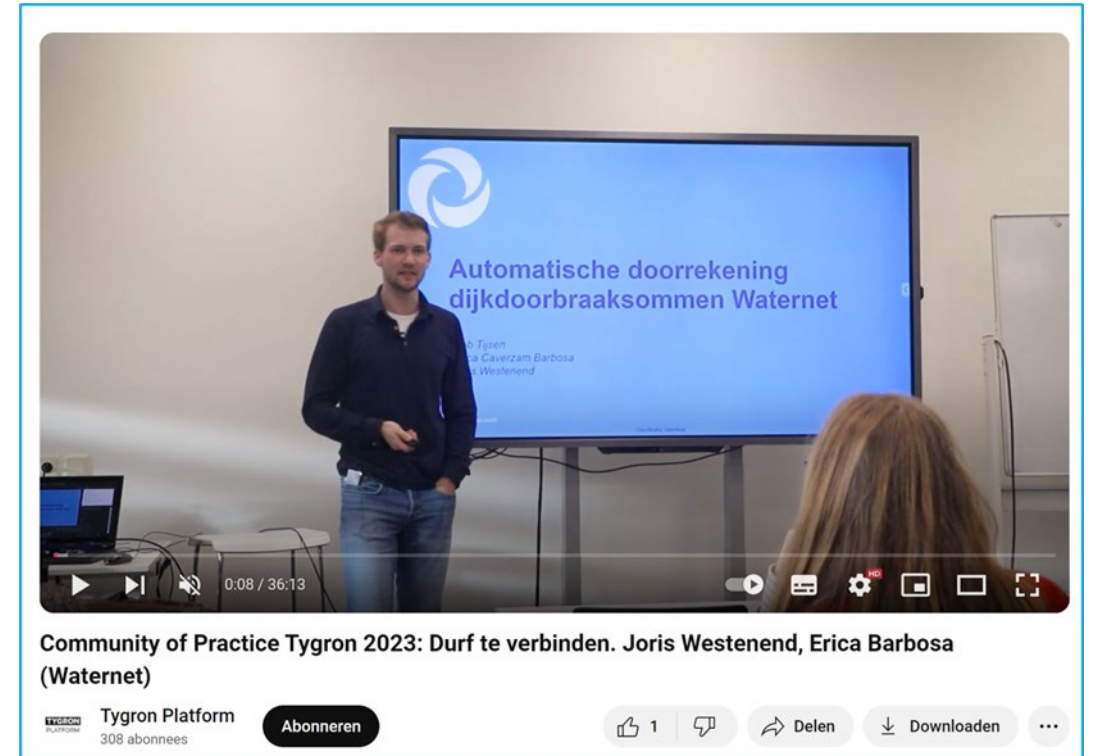
2000 doorbraaklocaties

Korte terugblik

Tygron CoP 2023

- Aannames, uitgangspunten en opzet
- Gevoeligheidsanalyse (kritieke stroomsnelheid, afstand meetpunt, gridcelgrootte)

[Link](#)



Inhoud – hoe zijn we verder gegaan?

- Samenwerking met Hoogheemraadschap van Rijnland
- Breslocaties en metadata
- Voorbeeld van ons template-project
- Samenwerking met Tygron
- Verwerking van de resultaten
- Voorbeeld van resultaat in landelijke database

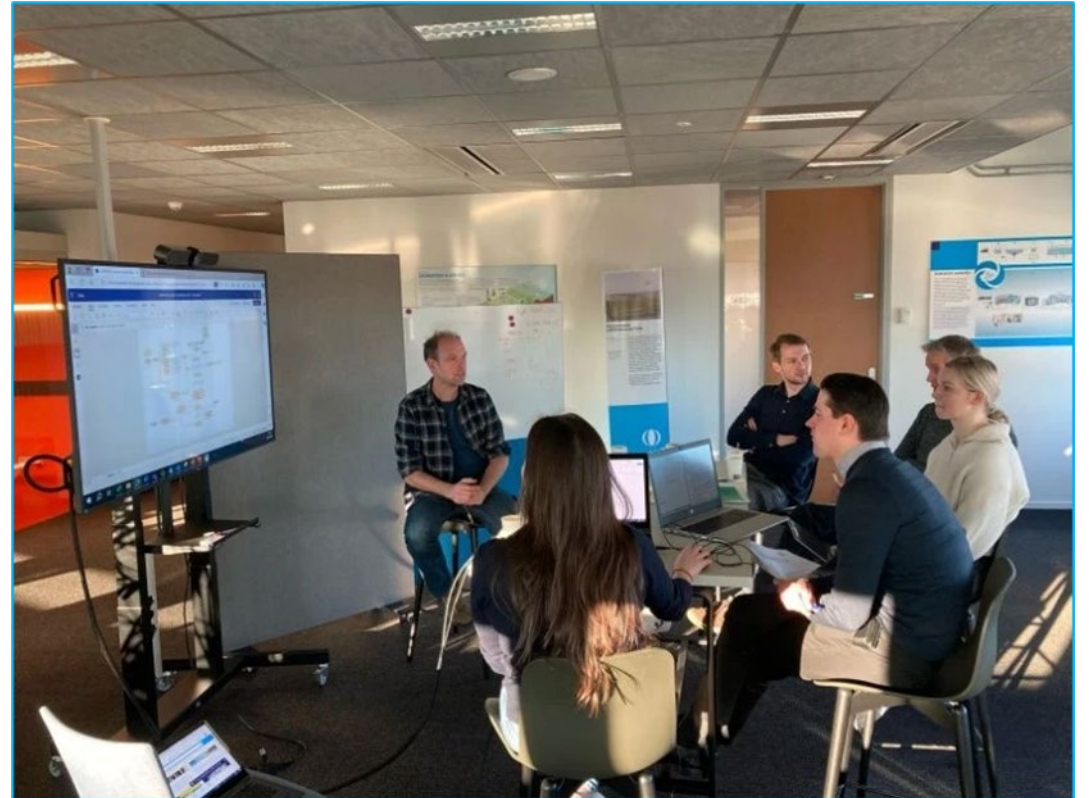
Samenwerking met Rijnland

AGV en Rijnland leveren deels aan zelfde provincie informatie aan

- Afstemming over uitgangspunten
- Afstemming over scripts & templates

Voordeel: kennisdeling

Resultaat: uniformere resultaten in landelijke database



Breslocaties en metadata

Voorbeeld Tygron

Samenwerking met Tygron

Regelmatig contact met Tygron

- Meerdere locaties doorrekenen binnen 1 bestaand project
- Triggers
- Bepaling hoogte bres in bebouwd gebied
- Scheduled updates uitvoeren met script

```
18     "generate" : false,  
19  
20     "recalculate_reset_sequence" : [true, false],  
21     "recalculate_scheduled_sequence" : [false, true],  
22     "recalculate_overlay_active" : ["Hoogtekaart", null],  
23     "timeout_in_seconds" : 1200,  
24     "scheduled_timeout_in_seconds" : 2400,
```

Geautomatiseerd keringen doorrekenen en data verwerken

Verwerkingscripts

- Resampling, aanvullen metadata, stuur met API resultaten naar LDO

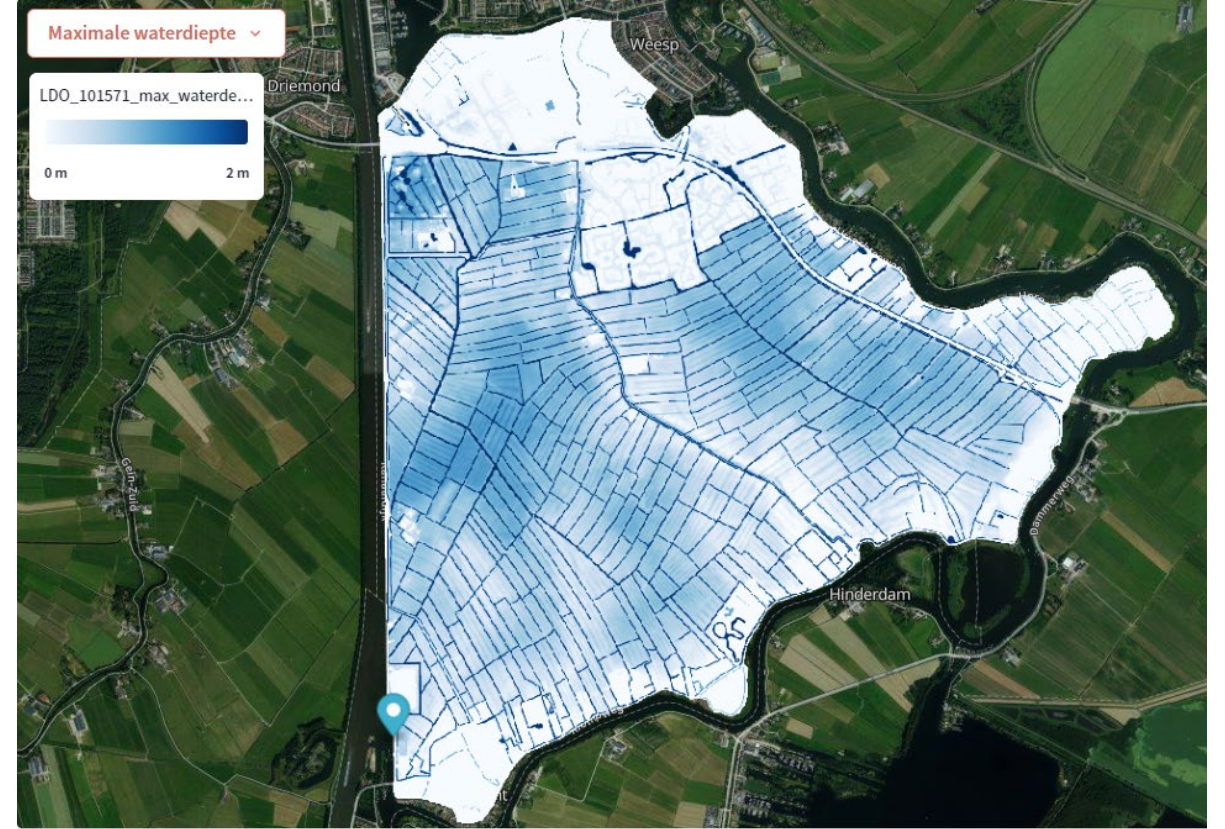
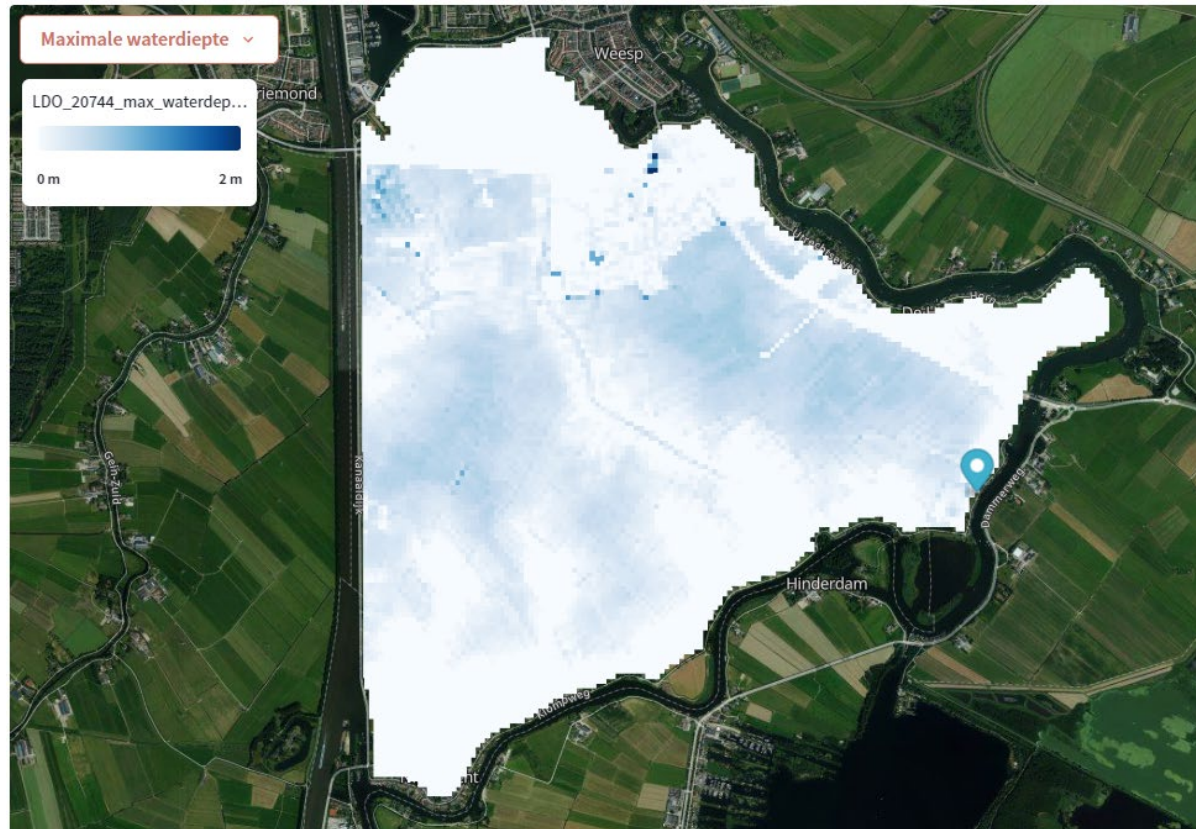
```
14 ##Definieer run-naam
15 run_name = "run01"
16
17 ##Definieer gewenste resolutie
18 xres = 5.0
19 yres = 5.0
20
21 ##Laad csv met namen tijdstappen in Tygron en voor LDO
22 rename_table = pandas.read_excel("rename_table.xlsx", dtype={'Oud Nummer': str, 'Nieuw Nummer': str})
23
24 ##Maak lijst aan en vul met tiff bestanden
25 tiffs_lijst = []
26 for file in glob.glob("*.tiff"):
27     tiffs_lijst.append(file)
28
29 ##Loop over de tiffs
30 for i in range(0,(len(tiffs_lijst))):
31
32     ##Resample
33     with rasterio.open((tiffs_lijst[i])) as dataset:
34
35         scale_factor_x = dataset.res[0]/xres
36         scale_factor_y = dataset.res[1]/yres
37
38         profile = dataset.profile.copy()
39
40         data = dataset.read(
41             out_shape=(
42                 dataset.count,
43                 int(dataset.height * scale_factor_y),
44                 int( dataset.width * scale_factor_x)
45             ),
46             resampling=Resampling.nearest
47         )
48
49         transform = dataset.transform * dataset.transform.scale(
50             (dataset.width / data.shape[-1]),
51             (dataset.height / data.shape[-2])
52         )
53         profile.update({"height": data.shape[-2],
54                       "width": data.shape[-1],
55                       "transform": transform})
56
```

```
1 """Deze module leest een van Tygron ontvangen HTML bestand in
2 en neemt de waarden daaruit over in onze eigen tabellen.
3 Deze code is gebaseerd op het notebook
4 'Python\Scripts_hydrologie\Output_HTML\Output_Tygron_HTML.ipynb' van Erica
5 """
6
7 import re
8
9 testHtml = r"C:\Users\Adm_rijn15\Documents\Repos\geo-fme-tygron\Projectdata\
10
11 def fme_connectietest():
12     """Om alvast de verbinding met dit script te testen in FME"""
13     print ("***** tygron_html_caller.py gevonden *****")
14
15
16 def extract_values_from_html(html_content):
17     # Extract values of each parameter
18     patterns = {
19         "MAX BRESBREEDTE": r"MAX BRESBREEDTE \(\m\): (\d+\.\d+)",
20         "MAX BRESDEBIET": r"MAX BRESDEBIET \(\m3/s\): (\d+\.\d+)",
21         "TOTAAL DEBIET": r"TOTAAL DEBIET \(\m3\): (\d+\.\d+)",
22         "MAX WATERSTAND": r"MAX WATERSTAND \(\mMAP\): (-?\d+\.\d+)"
23     }
24     values = {}
25     for key, pattern in patterns.items():
26         match = re.search(pattern, html_content)
27         if match:
28             values[key] = float(match.group(1))
29         else:
30             values[key] = None
31     return values
32
33
34 def main(html_file):
35     values={}
36     with open(html_file, 'r') as f:
37         html_content = f.read()
38         values = extract_values_from_html(html_content)
39     print(values)
40     return(values)
41
42
43 if __name__ == '__main__':
44     main(testHtml)
```

Voorbeeld resultaten LDO

Landelijke Database Overstromingsrisico's

Verskil oud en nieuw





Vragen?

E-mail: joris.westenend@waternet.nl