



HYDROLIB & NHI

Volgende stap voor een ontwikkelcommunity

Ruben Dahm, Arthur van Dam, Rinske Hutten

NHI-dag, 19 januari 2023

Wat is HYDROLIB?

- Een initiatief van 13 organisaties (waterschappen, adviesbureaus en kennisinstellingen)
- Een softwarebibliotheek (set Python algoritmes en tools) voor interactie met de D-HYDRO Suite 1D2D voor oppervlaktewater en de HyDAMO-datastandaard
- Een gezamenlijke stap die aansluit op een behoefte van automatisering en reproduceerbaarheid en daarmee de kwaliteit van het modelleringsproces te vergroten
- Daarmee geven de tools van HYDROLIB vertrouwen want gestandaardiseerd en gecontroleerd

HYDROLIB: herbruikbare software tools (vnl. Python)

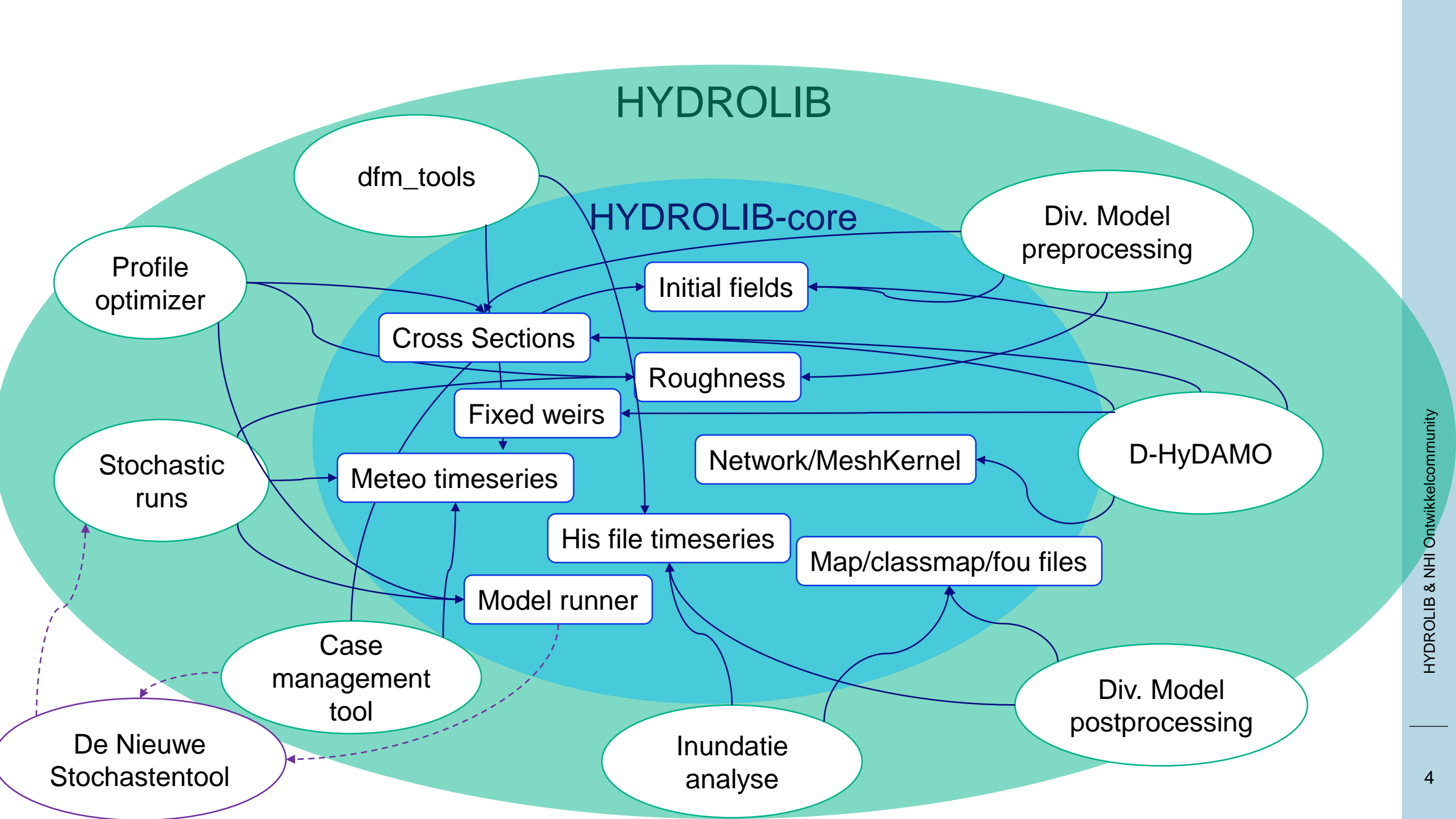
Core scripts & tools voor interactie met D-HYDRO

- Centraal beschikbaar, online.
- Ondersteund, Gedocumenteerd, en onderhouden, ook voor nieuwe releases.
- Open source, met “markt-compatible” licentiekeuze.

D-HYDRO gebruikers kunnen **concentreren op hun specialistische/afgeleide hydrologische tools** en diensten.

Door bovenop de core scripts te bouwen.



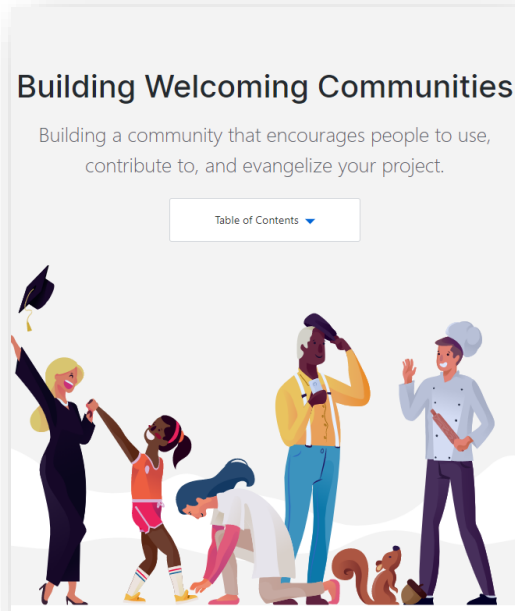


Fase 1: TKI project HYDROLIB: Winter 2021 – zomer 2022: focus op 1D2D waterschapstoepassingen.



Het begin: samen bedenken hoe samenwerken werkt

Bijvoorbeeld: de After-lunch mini-lectures (2021)



HYDROLIB als 'Open source community'

26 april

		
Type	Permissive	Copyleft
Provides copyright protection	✓ TRUE	✓ TRUE
Can be used in commercial applications	✓ TRUE	✓ TRUE
Provides an explicit patent license	✗ FALSE	✗ FALSE
Can be used in proprietary (closed source) projects	✓ TRUE	✗ FALSE

Licenties en IP

05 mei



Samen ontwikkelen op een platform

12 mei



Documentatie

19 mei

Documentatie

Gebruikersdocumentatie

HYDROLIB documentation

Home About How-to Tools Reference Changelog

Tools

- D-HyDAMO plugin
- Profile Optimizer
 - API reference
- Inundation Toolbox
- Case Management Tools
- ARCADIS Tools
- HydroMT-Delft3D FM plugin

Introduction

The Profile Optimizer is a py (D-Flow FM) model, a part of the bottom width of a single width is optimised to reach

The Profile Optimizer can be coefficient could be change could test for an desired wa

Usage

Tutorials/trainingen

```
plt.show()
```

For finalizing the FM-model, we also need the coupling to the other modules. Therefore, we will do that first.

Add an RTC model

RTC contains many different options. Three are now implemented in D-HyDAMO:

- a PID controller (crest level is determined by water level at an observation point);
- a time controller (a time series of crest level is provided);
- the possibility for the users to provide their own XML-files for more complex cases. Depending on the complexity, the integration might not yet work for all cases.

First, initialize a DRTCModel-object. The input is hydamo (for the data), fm (for the time settings), a path where the model will be created (typically an 'rtc' subfolder), a timestep (default 60 seconds) and, optionally, a folder where the user can put 'custom' XML code that will be integrated in the RTC-model. These files will be parsed now and be integrated later.

```
if RTC:
    drtcmodel = DRTCModel(
        hydamo,
        fm,
        output_path=output_path,
        rtc_timestep=60.0,
        complex_controllers_folder=data_path / "complex_controllers"
    )
```

If PID controllers are present, they need settings that are not included in the HyDAMO DAMO2.2 data. We define those in a dictionary. They can be specified for each structure - in that case the key of the dictionary should match the key in the HyDAMO DAMO2.2 'sturing'-object. If no

API documentatie

FrictionModel ([INIModel](#)) pydantic-model

The overall friction model that contains the content

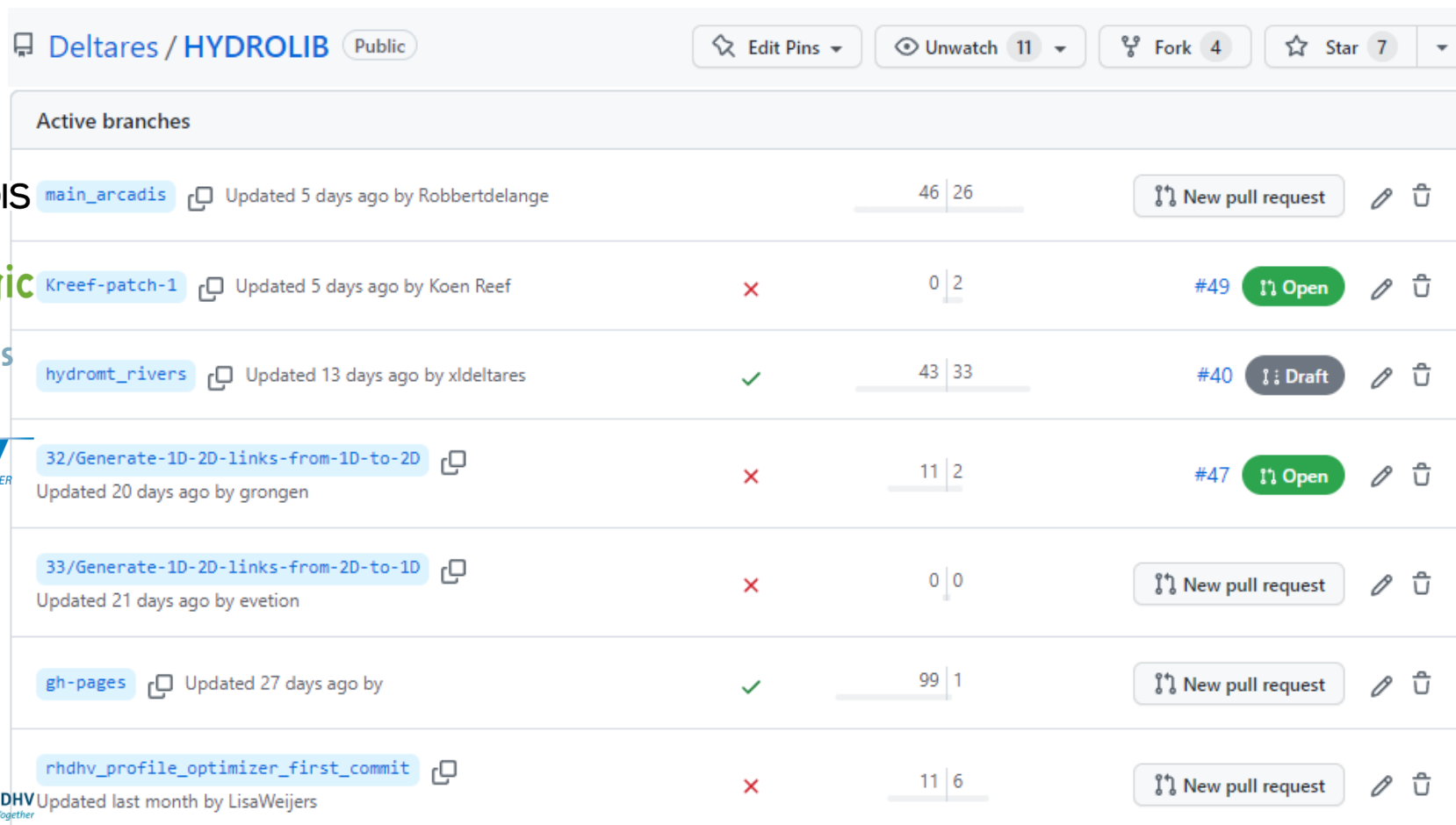
This model is typically referenced under a [FMModel](#)

Attributes:

Name	Type	Description
<code>general</code>	<code>FrictGeneral</code>	[General] block
<code>global_</code>	<code>List[FrictGlobal]</code>	Definitions of [G
<code>branch</code>	<code>List[FrictBranch]</code>	Definitions of [B

HYDROLIB bijdrages

- Bureau's hebben pilot- en tool-scripts ontwikkeld, en Deltares werkt aan de kern.
- Iedereen werkt in code-branches (vroeg delen met elkaar, maar wel een stabiele hoofdlijn houden).



The screenshot shows the GitHub repository 'Deltares / HYDROLIB' with the following active branches and pull requests:

Branch	Updated	Status	Commits	PRs	PR Title
main_arcadis	Updated 5 days ago by Robbertdelange	✓	46	26	New pull request
Kreef-patch-1	Updated 5 days ago by Koen Reef	✗	0	2	#49 Open
hydromt_rivers	Updated 13 days ago by xldeltares	✓	43	33	#40 Draft
32/Generate-1D-2D-links-from-1D-to-2D	Updated 20 days ago by grongen	✗	11	2	#47 Open
33/Generate-1D-2D-links-from-2D-to-1D	Updated 21 days ago by evetion	✗	0	0	New pull request
gh-pages	Updated 27 days ago by	✓	99	1	New pull request
rhdhv_profile_optimizer_first_commit	Updated last month by LisaWeijers	✗	11	6	New pull request



HYDROLIB tools op dit moment

Projectresultaten: hydrolib-core

hydrolib-core releases met support voor vrijwel alle invoer van D-HYDRO Suite 1D2D (en uitvoer komt).



Search projects

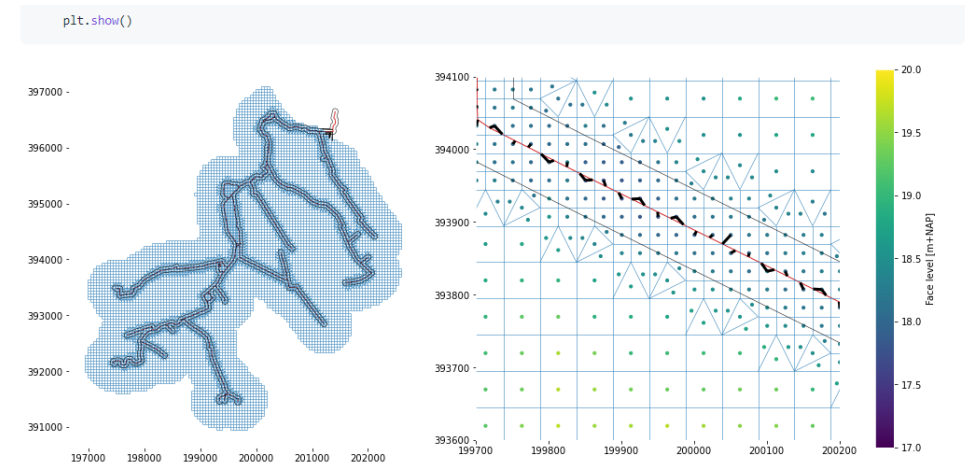
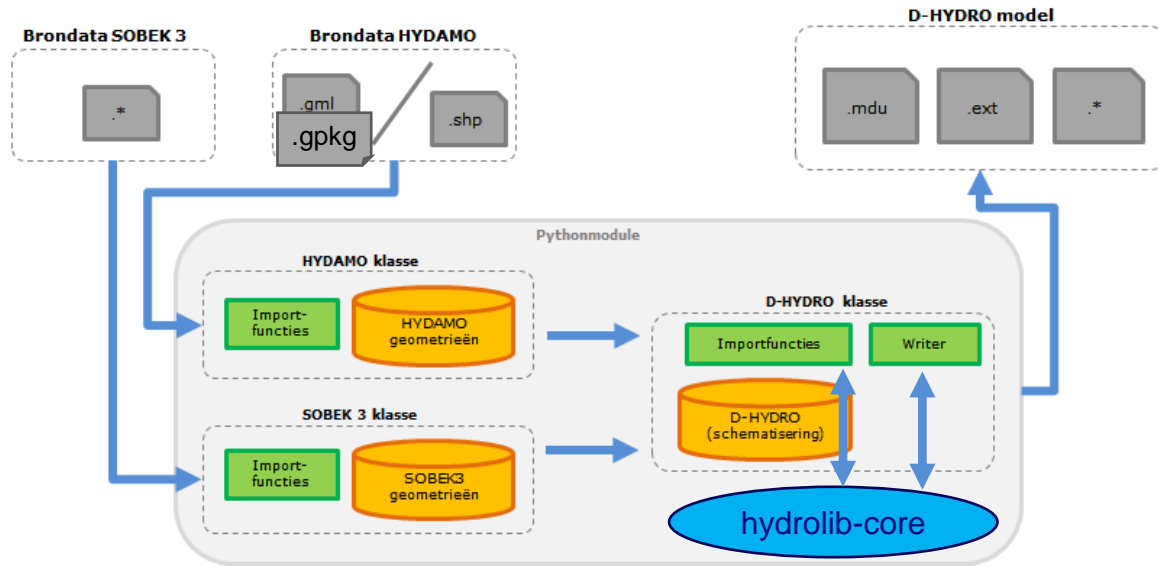
hydrolib-core 0.3.0

```
pip install hydrolib-core
```

“Als je eenmaal weet hoe je het gebruikt, werkt het best heel mooi.”

“Mijn code is er een stuk korter door geworden.”

D-HyDAMO: modelgeneratie HyDAMO → D-HYDRO



For finalizing the FM-model, we also need the coupling to the other modules. Therefore, we will do that first.

Add an RTC model

RTC contains many different options. Three are now implemented in D-HyDAMO:

- a PID controller (crest level is determined by water level at an observation point);
- a time controller (a time series of crest level is provided);
- the possibility for the users to provide their own XML-files for more complex cases. Depending on the complexity, the integration might not yet work for all cases.

First, initialize a DRTCModel-object. The input is `hydamo` (for the data), `fm` (for the time settings), a path where the model will be created (typically an 'rtc' subfolder), a timestep (default 60 seconds) and, optionally, a folder where the user can put 'custom' XML code that will be integrated in the RTC-model. These files will be parsed now and be integrated later.

```
if RTC:
    drtcmodel = DRTCModel(
        hydamo,
        fm,
        output_path=output_path,
        rtc_timestep=60.0,
        complex_controllers_folder=data_path / "complex_controllers"
    )
```

If PID controllers are present, they need settings that are not included in the HyDAMO DAMO2.2 data. We define those in a dictionary. They can be specified for each structure - in that case the key of the dictionary should match the key in the HyDAMO DAMO2.2 'sturing'-object. If no





Profile Optimizer DHydro - Jupyter notebook



Introductie

De **Profile Optimizer** is een Python-tool waarin het optimaliseren van profielen voor D-Hydro modellen geautomatiseerd is. Gebaseerd op een bestaand D-Hydro model, zal een deel van het systeem worden gewijzigd om een geoptimaliseerd bodembreedte van één profiel van één tak te optimaliseren aan de hand van de hu

geoptimaliseerd, zodat een gewenste stroomsnelheid ontstaat bij de gekozen profi

Contact

De Profile Optimizer is onderdeel van HYDROLIB, een open source community voor Hydrolib voor meer informatie: <https://github.com/Deltares/HYDROLIB>

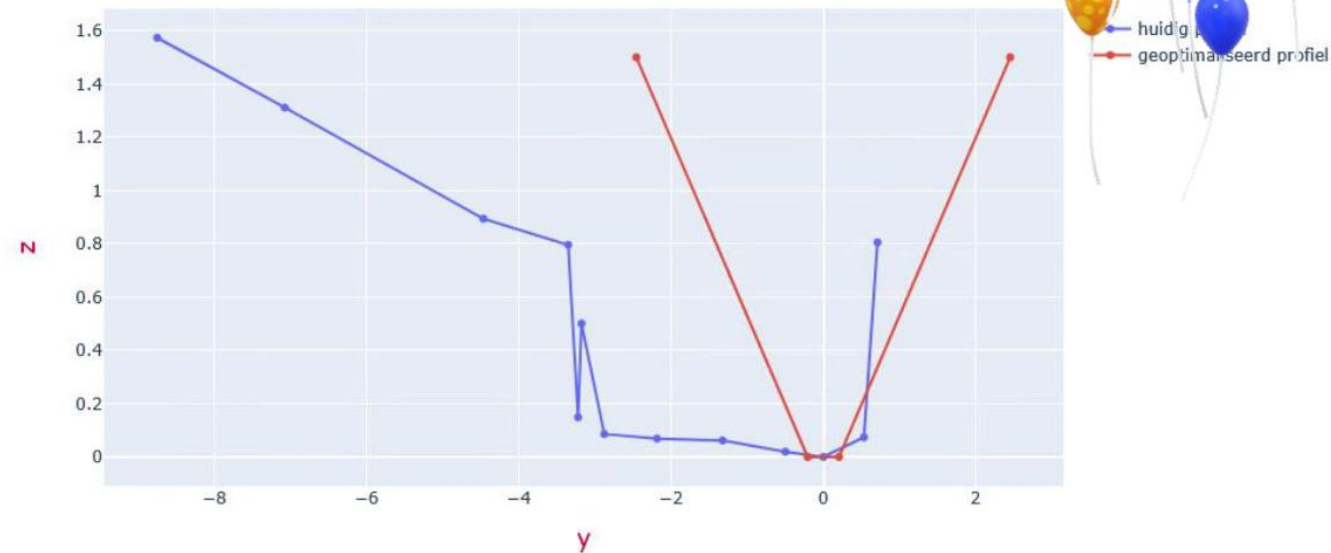
De Profile Optimizer is ontwikkeld door Royal HaskoningDHV:

- rineke.hulsman@rhdhv.com
- lisa.weijers@rhdhv.com
- galerie.demetriades@rhdhv.com

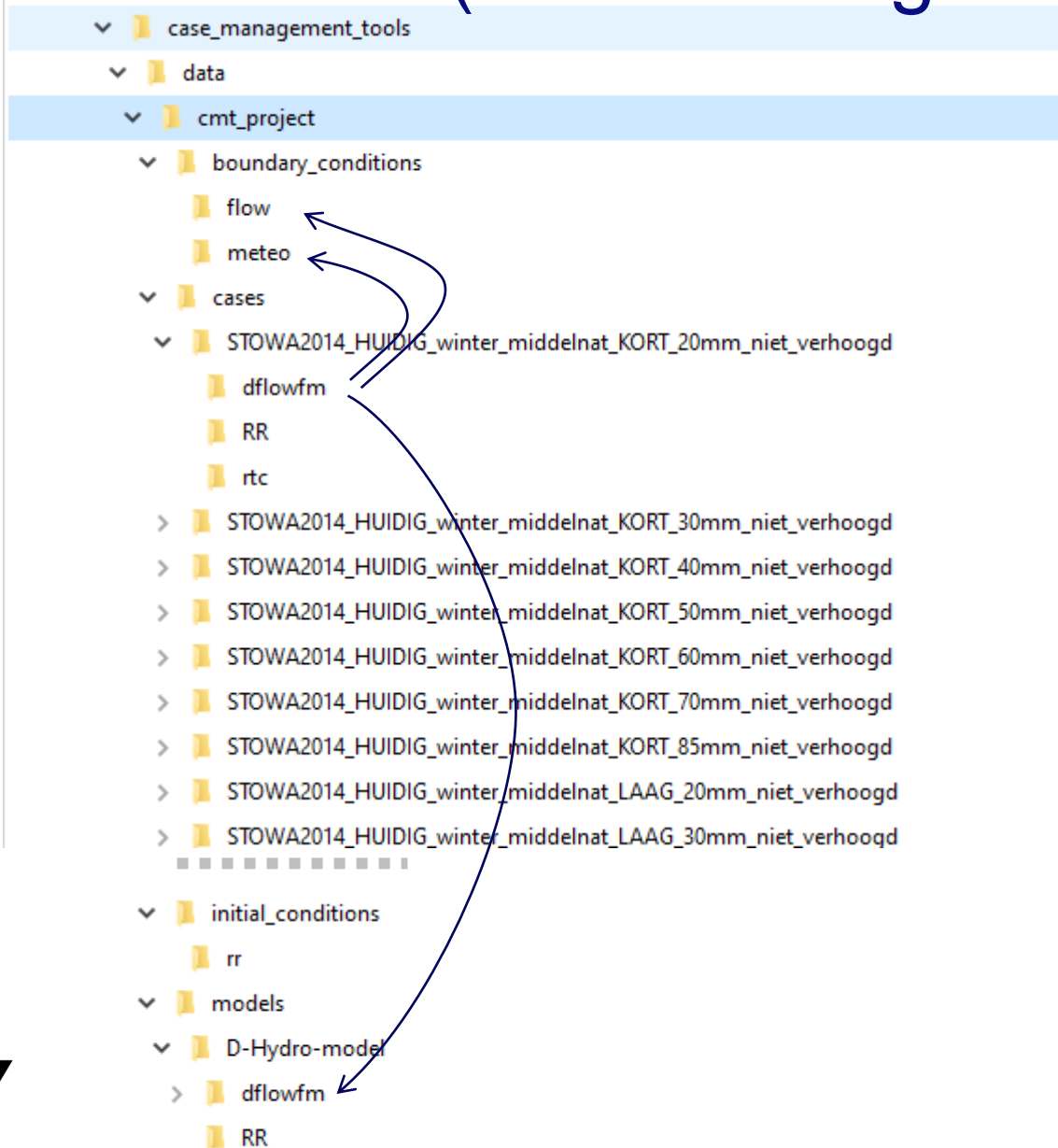
Content

- [Stap 0: Klaarzetten input](#)
- [Stap 1: Kies optimalisatie gebied/locatie](#)

DP_433814



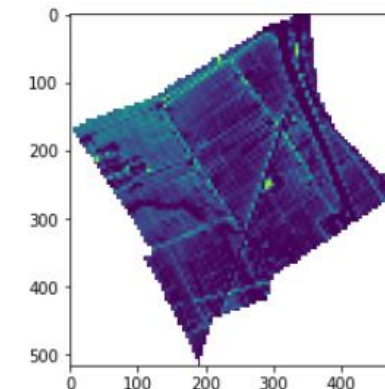
Model runs (case management), Postprocessing



```
In [5]: # Load mesh coordinates and data from netCDF
node_data = load_fou_data(input_file_path, variable)
```

```
In [6]: # convert to raster and save as tiff
_, _, grid_data = mesh_to_tiff(
    node_data,
    input_file_path,
    output_file_path + r"\fou.tiff",
    resolution,
    distance_tol,
    interpolation=interpolation,
)
```

```
In [7]: # plot
plt.imshow(grid_data[-1::-1, :])
plt.show()
```





HYDROLIB Ontwikkelcommunity

Stimuleren codedeling: belonen in uitvragen?

Veel partijen (m.n. adviesbureaus)

- staan open voor code delen,
- of zijn in elk geval niet tegen.

Nodig is dan wel:

- Richtlijnen
- Dit stimuleren vanuit opdrachtgevers.



Richtlijnen om HYDROLIB Python scripts conform standaard op te leveren

Versie 0.4

De HYDROLIB verzameling van Python tools voor hydrodynamische workflows staat open voor toevoegen van nieuw ontwikkelde scripts en tools. Dit versterkt de herbruikbaarheid en inzetbaarheid van deze code in o.a. modelstudies. De code van HYDROLIB leeft op GitHub: <https://github.com/Deltares/HYDROLIB>.

Om code toe te voegen moet gebruik worden gemaakt van het versiebeheersysteem *git*. Nieuwe of aangepaste code ontwikkelt men daarbij in een *branch*: een kopie van de hoofdlijn waarin ontwikkelingen zonder problemen voor andere gebruikers kunnen worden geïmplementeerd. Om de code naar de hoofdlijn te verplaatsen maken we gebruik van *zgn. pull requests*: een aanvraag om code op te nemen in de hoofdlijn. In de hoofdlijn kan na elke geaccepteerde *pull request* een nieuwe versie¹ gepubliceerd worden, die voor iedereen ook met *pip* te installeren is. Voordat een *pull request* geaccepteerd kan worden, moet de nieuw toe te voegen code voldoen aan een aantal eisen in dit document beschreven als richtlijn. Hiermee garanderen we de kwaliteit en bruikbaarheid van HYDROLIB, nu en in de toekomst.

Dit document is een gezamenlijk product van de 13 organisaties die HYDROLIB geïnitieerd hebben. De richtlijn bestaat uit een checklist die doorlopen kan worden door een partij die een tool (set aan scripts) of enkele losse scripts wil opleveren in HYDROLIB. De checklist bestaat uit 3 niveaus: generiek, vereist, aanbeveling:

- Generiek:
 - o Akkoord met licentie
 - o Intellectueel eigendom
 - o Aangemeld als *contributor*
- Vereist:
 - o Generieke scripts
 - o Markers voor expliciete keuzes
 - o Environment
 - o Documentatie:

Ontwikkelcommunity: we hebben wat lessen geleerd

- “Dare to share” moet wel commerciële activiteiten toestaan (anders niet interessant voor bureaus). **Verschillende licenties** voor de core (MIT) en de 3rd party tools (LGPL of zelfs gesloten).
- Leg **duidelijke keuzes** neer voor: code conventies (Black), documentatie (Google-style docstrings) en voorkeurs-Python packages (pathlib, numpy, xarray).
- Zonder coördinatie wordt code wordt wel **wel gedeeld, maar niet hergebruikt**.
- Samenwerkdagen **op kantoor** zijn geweldig nuttig.
- Branchen, mergen en elkaars code reviewen op **GitHub werkt heel fijn**.

Opgeleverd, wat nu?

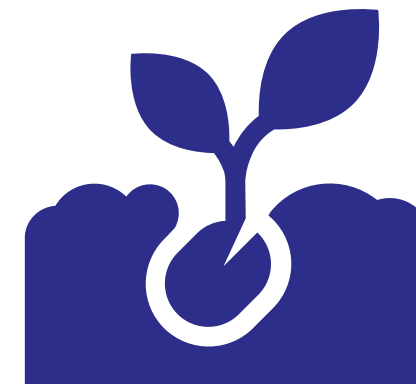


beheren



community

groei





Lange termijn (na 2023)

Beheer en Ontwikkeling (voorstel pilot 2023)
1^e ideeën voorgelegd Stuurgroep (SRLM)
Verkenning van rolverdeling



Korte termijn (in 2023)

Doorstarten


- Laagdrempelig (introductietrainingen)
- HYDROLIB-NHI Labs (samenwerkdagen)
- Train de Reviewer
- Maandelijks Stand-Up



Meer weten?

  deltares.github.io/HYDROLIB/

 hydrolib@deltares.nl

HYDR**LIB**

