

interpolatie tussen twee coördinaten (bodembreedte 1 + stroomsnelheid 1 en bodembreedte 2 + stroomsnelheid 2) naar een gewenste stroomsnelheid, de juiste bodembreedte wordt gevonden. Deze relatie is niet per definitie lineair. Het is een redelijke benadering, die beter is wanneer er kleine ruimte zit tussen de coördinaten.

### 3.4 Toepasbaarheid

Op dit moment is de profile optimizer getest op een 1D FM model. Andere modules zouden geen impact moeten hebben op het functioneren van het model, maar is daar niet op getest. Daarnaast dienen de profielen gedefinieerd te zijn als een YZ-profiel. Het te optimaliseren gebied dient niet te groot te zijn, omdat alle profielen in het geselecteerde gebied hetzelfde profiel krijgen en de stroomsnelheid op één locatie gecheckt wordt. De bodemhoogte van het huidige profiel blijft wel behouden. Daarnaast krijgt het nieuwe profiel hetzelfde talud aan beide zijden. De profile optimizer werkt op dit moment met één constraint; namelijk de stroomsnelheid. De waterdiepte wordt wel gepresenteerd in de Jupyter Notebook voor de gebruiker ter informatie.

### 3.5 Verbeterpunten / aanbevelingen

Er zijn een aantal verbeterpunten gedefinieerd op basis van dit eerste concept:

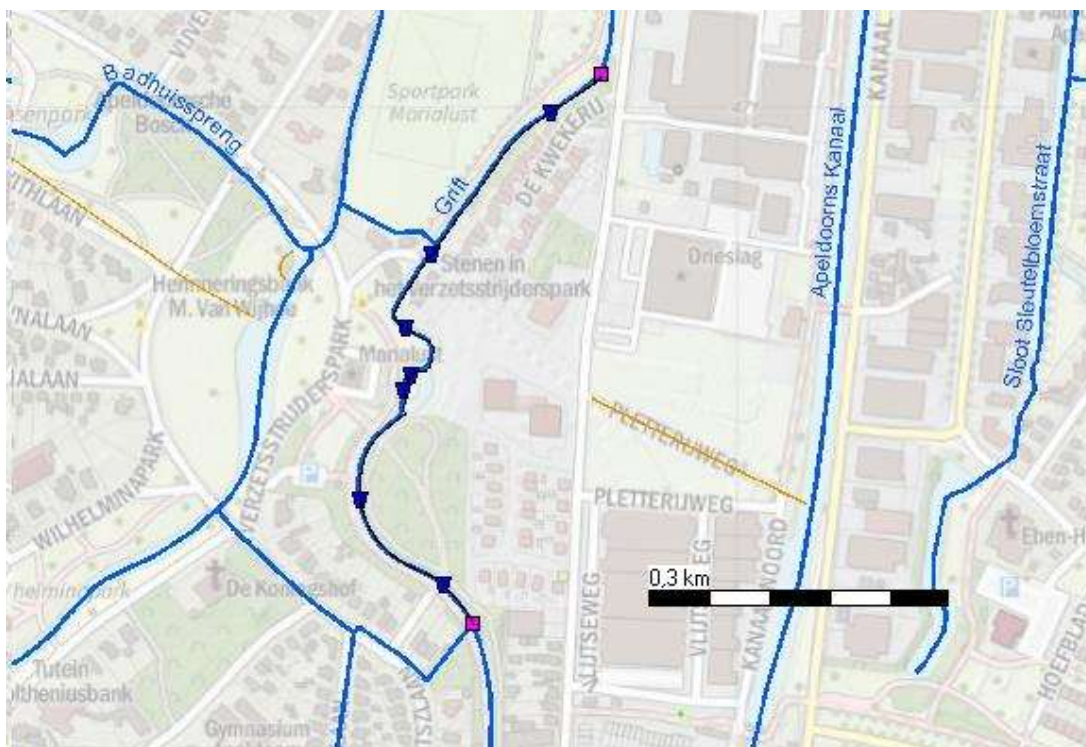
- Meerdere profiel optimalisaties (optimalisatie per profiel).
- Stroomsnelheid toetsen op meerdere locaties (constraint).
- Waterdiepte meenemen in de toetsing voor optimale bodembreedte (extra constraint).
- Controle uitvoeren op waterstand op een opgegeven locatie.
- Toepasbaar maken op vertakkingen (hiervoor is de eerst genoemde verbetering noodzakelijk).
- Uitvoeren / toetsen op verschillende afvoersituaties.
- Functies verder generaliseren.
- Functies gebruik laten maken van MDU in plaats van handmatige padnamen voor modelementen.
- In de initialisatie al modelresultaten van het referentiemodel gebruiken in plaats van handmatig kenmerken opgeven.

## 4 Toepassing pilotgebied

### Aanpak

Om een idee te krijgen van de stroomsnelheden in de Grift bij Marialust is er een 1D-model in SOBEK gemaakt waarin de ingemeten dwarsprofielen zijn opgenomen. De berekening gaat uit van een laminaire stroming waarbij een gemiddelde stroomsnelheid over het profiel wordt berekend. Dit is een versimpeling van de werkelijkheid en geeft bij benadering een beeld van de stroomsnelheden ter plekke.

In de onderstaande afbeelding zijn de locaties van de ingemeten dwarsprofielen en de boven- en benedenstroomse randvoorwaarden weergegeven.

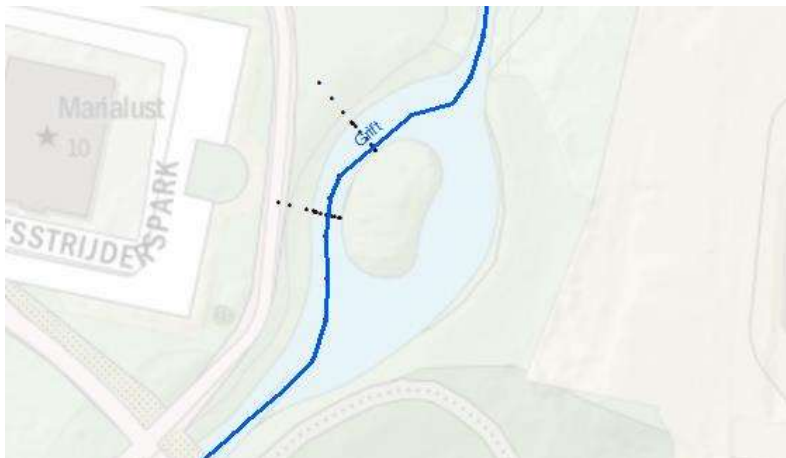


Figuur 4-1: schematisatie van het SOBEK model

De volgende uitgangspunten zijn gehanteerd:

- Bovenstroomse randvoorwaarde:  $Q = 0,369 \text{ m}^3/\text{s}$  maatgevende afvoer (vanuit GIS-laag beleidsharmonisatie).
- Benedenstroomse randvoorwaarde: kruinhoogte 11,6 m NAP van stuw KST-4527.
- Wandruwheid (begroeiing): Bos en Bijkerk 25.

De dwarsprofielen zijn op 14-9-2008 ingemeten op de twee dwarsprofielen op de onderstaande afbeeldingen, die zijn op 20-4-2018 ingemeten. Voor het oostelijke deel van de vertakking zijn geen gegevens beschikbaar van de afmetingen van het dwarsprofiel. Hiervoor heb is de eenvoudige aanname gedaan dat het dwarsprofiel aan de andere kant van het eilandje gelijk is aan het deel dat wel is ingemeten.



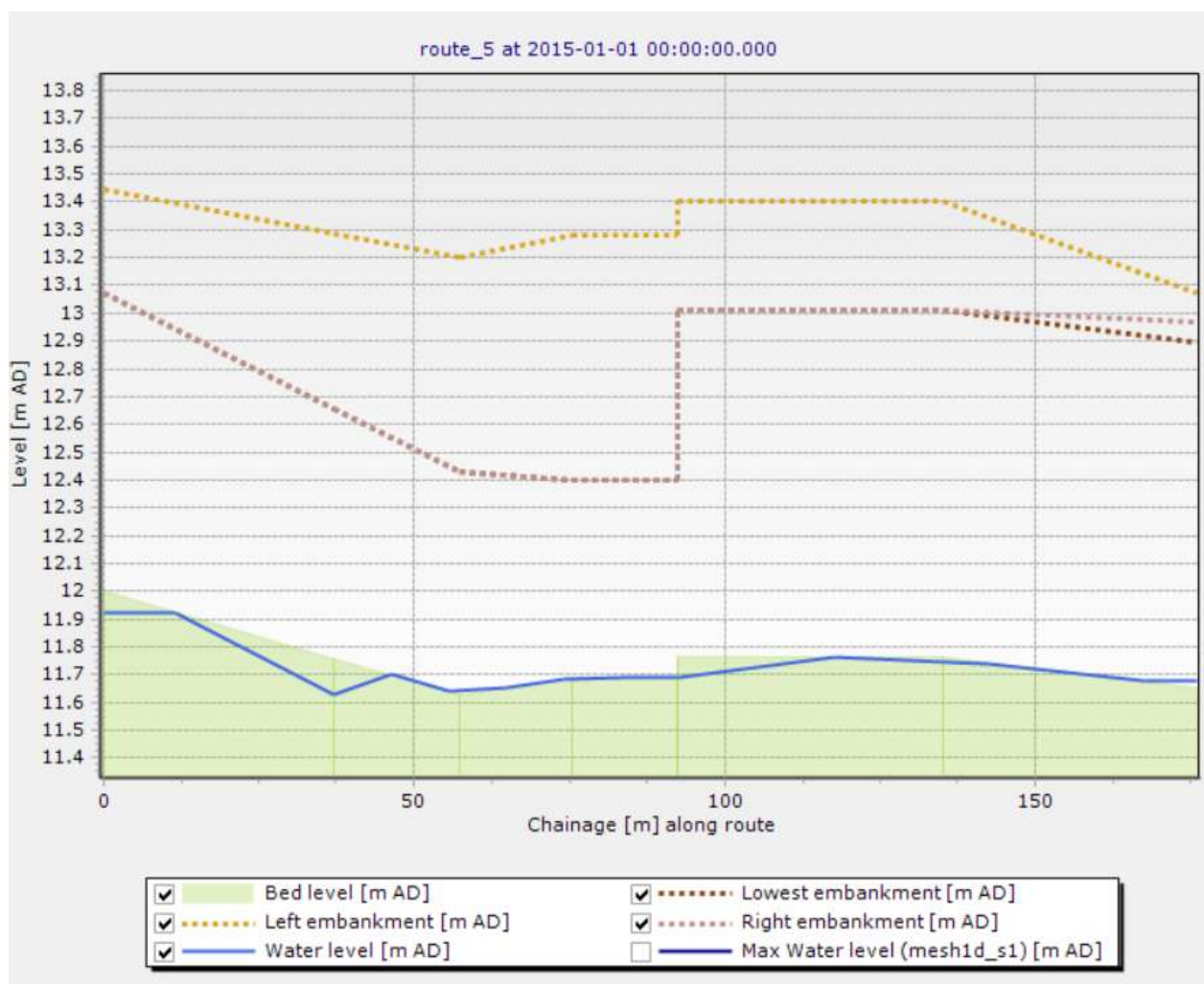
Figuur 4-2: Inmeting dwarsprofielen t.p.v. Marialust.

### Resultaten en conclusies

Het SOBEK-model is omgezet naar een D-HYDRO model en vervolgens is de huidige situatie en de situatie met de afdamming van het westelijke deel langs het eiland doorgerekend. In de onderstaande figuren zijn de berekende stroomsnelheden weergegeven.



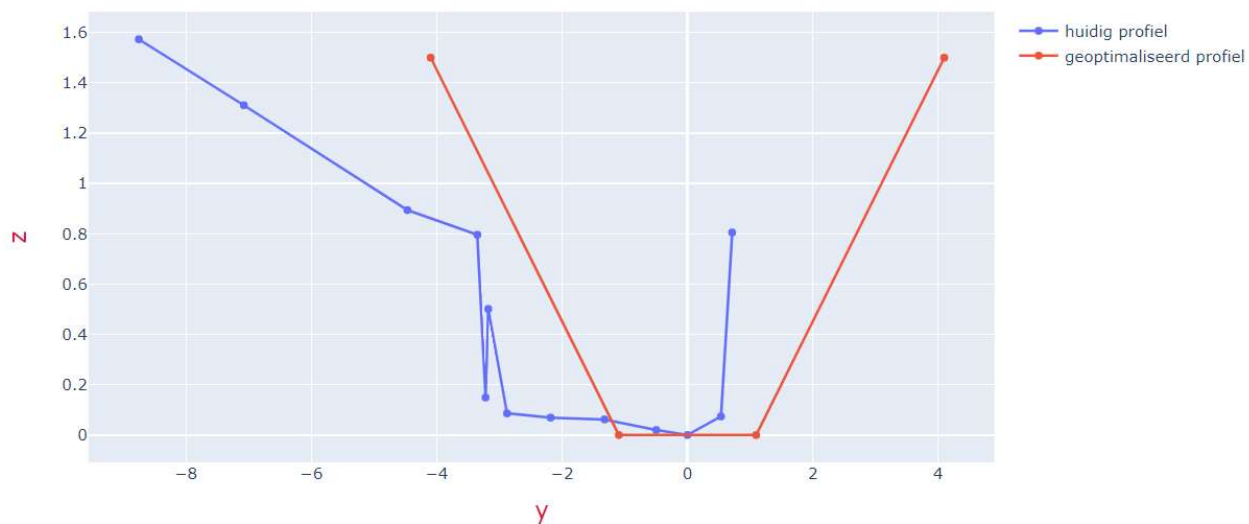
De stroomsnelheid van het water wordt voornamelijk bepaald door enerzijds de grootte van het dwarsprofiel (natte oppervlak) en anderzijds het aanwezige verhang in de watergang. In de huidige situatie zijn de stroomsnelheden aan beide zijden van het eiland laag ( $< 10\text{ cm/s}$ ). Boven- en benedenstrooms van het Marialust zijn de stroomsnelheden al snel een factor 3 à 4 groter. In de onderstaande afbeelding (Figuur 4-3) is het verhang in het systeem weergegeven in de huidige situatie.



Figuur 4-3: Zijaanzicht, Marialust ligt tussen 50 en 100 meter chainage.

Op het moment dat het westelijke deel van het eiland wordt afgedamd dan wordt de stroomsnelheid in het oostelijke deel aanzienlijk beter en neemt de kans en de hoeveelheid sedimentatie van slib daarmee af. Of er in het geheel niet meer gebaggerd hoeft te worden kan niet uitgesloten worden. Om de stroomsnelheid verder te vergroten zal de bodembreedte moeten worden aangepast. Met behulp van de profile optimizer is verkend wat de gewenste bodembreedte is om aan de gewenste stroomsnelheid te voldoen. In de figuur 4-4 is het huidige profiel en het geoptimaliseerde profiel getoond. Zoals was te verwachten is het geoptimaliseerde profiel aanzienlijk kleiner ( $B=2,18\text{ m}$ ) dan het bestaande profiel ( $B=3,76\text{ m}$ ).

DP\_433814



Figuur 4-4: Geoptimaliseerde profiel (rood) en huidige profiel (blauw).

De berekende waterdiepte is 0,57 meter. Een dynamische berekening zou vervolgens moeten aantonen of deze bodembreedte bij hogere afvoeren tot knelpunten zou kunnen leiden.



## 5 Technische beschrijving profile optimizer

De profile optimizer is te gebruiken via de Jupyter notebook. De Jupyter Notebook is voorzien van uitleg van iedere stap en importeert de verschillende onderliggende python functies. De technische beschrijving bestaat uit de volgende stappen: uitleg opbouw Python scripts, uitleg workflow en een overzicht van de aannames. Deze beschrijving behoort bij versie 0.1.2.

### 5.1 Opbouw Python scripts

Profile optimizer versie 0.1.2 bestaat uit vier python scripts:

- **geometry.py:** Bevat algemene functies voor het genereren van geometrische objecten van het D-HYDRO Suite 1D2D model: de ligging van waterlopen en profielen in het model;
- **optimizer.py:** Bevat de kern van de optimalisatieprocedure en het iteratief rekenen met D-HYDRO Suite 1D2D;
- **postprocessing.py:** Bevat postprocessing functies zoals het uitlezen van D-HYDRO Suite 1D2D resultaten en het plotten van profielen;
- **preprocessing.py:** Bevat functies voor de initialisatie van de profile optimizer (initiële bodembreedte, bandbreedte zoekruimte).

In de volgende hoofdstukken worden deze scripts toegelicht.

#### 5.1.1 geometry.py

In **geometry.py** zitten drie functies: `create_branches`, `create_crosssections` en `select_crosssection locations`. De functies maken gebruik van de `netFile`<sup>1</sup> en `CrossLocFile`<sup>2</sup> bestand.

Vanuit de `netFile` bouwt **create\_branches** een geometrisch bestand van de waterlopen, het resultaat is een geodataframe (geopandas). Om deze geometrieën te maken worden eerst alle vertexen als punt aangemaakt, met de velden "network\_geom\_x<sup>3</sup>" & "network\_geom\_y" voor alle vertexen. De branches worden samengesteld met "network\_geom\_node\_count" voor het aantal vertexen per branch, "network\_branch\_id" voor de naam van de branch en tot slot "network\_edge\_length" voor de user length. Vervolgens worden de vertexen aan elkaar gekoppeld per branch, aan de hand van de "network\_geom\_node\_count".

Bijvoorbeeld: de eerste branch heeft als "network\_geom\_node\_count" 10, dan worden de eerste 10 vertexen samengevoegd tot een linestring. Als de volgende branch er 6 heeft, dan worden vertex 11 t/m 16 de linestring voor de tweede branch, enzovoort.

De locatie van profielen in D-HYDRO Suite 1D2D zijn gedefinieerd in de `CrossLocFile`. Deze locaties zijn relatief aan de lengte van de branch waar ze op zijn gelegen (chainage), daarom heeft de functie **create\_crosssections** als inputs: de `CrossLocFile` en een geodataframe van de branches (afkomstig uit `create_branches`). Met de `interpolate` functie van `shapely` kun je op een afstand X een punt op een lijn plaatsen. Vervolgens is het nog nodig om de afstand X (chainage) te verscalen op basis van de user length ("network\_edge\_length") van de branch. Het is namelijk mogelijk om een branch in D-HYDRO Suite 1D2D een andere lengte te geven dan de geometrische lengte. De chainage is op basis van de user length, en moet dus verschaald worden naar relatieve chainage op basis van de geometrische lengte van

<sup>1</sup> `netFile` is de benaming van de Network file in de MDU. Vaak heeft het bestand een naamgeving zoals `model_net.nc` of `model_network.nc`.

<sup>2</sup> `CrossLocFile` is de benaming in de MDU. Vaak heeft het bestand een naamgeving zoals `crosssection_locations.ini` of `crossloc.ini`

<sup>3</sup> Overal in de tekst waar "network\_" als prefix is gebruikt, zou ook "network1d\_" gebruikt worden. Dit verschilt per model, de `geometry` functies in de `profile optimizer` zoekt sinds versie 0.1.2 automatisch welke prefix is gebruikt.

een branch. Het resultaat van deze stap is een geodataframe (geopandas) met de locaties van alle crosssections in het model, met enkele andere kolommen waarvan de belangrijkste zijn: definitionid (naam crosssection definitie), branchid (naam van onderliggende branch) en ID (naam profiel-punt).

Bijvoorbeeld: branch A heeft een geometrische lengte van 10 meter, maar een user length van 15 meter. De chainage van een profiel op branch A is 7,5 meter. De vershalings factor is als volgt:

$$factor = \frac{\text{geometrische lengte}}{\text{user length}} = \frac{10}{15} = 0,6667$$

De relatieve chainage (scaled offset) is in dit geval dus  $7,5 \times 0,6667 = 5$  meter.

**select\_crosssections** is een laatste stap specifiek voor de profile optimizer, waarmee een selectie van de profielen gemaakt wordt. De profile optimizer past namelijk maar een deel van de profielen aan, op basis van een polygoon van het interessegebied. De input deze functie is dan ook een geodataframe van alle profielen en een shapefile. De output is een selectie van de geodataframe met profielen.

### 5.1.2 optimizer.py

In optimizer.py zit een class en een functie. De class is ProfileOptimizer, deze class beheert het aanmaken en doorrekenen van iteraties. Daarnaast is er een losse functie: find\_optimum, deze zoekt op basis van een set berekeningen (zoekruimte) wat de optimale oplossing is.

De iteraties worden geregeld door de **ProfileOptimizer** class. De class moet geïnitieerd worden met een aantal inputs:

- base\_model\_fn                      mdu van het bron-model;
- bat\_file                              een run.bat bestand waarmee de DIMR kan rekenen;
- work\_dir                              een map waar de iteraties tijdelijk opgeslagen worden;
- output\_dir                           een map waar de definitieve berekening wordt opgeslagen;
- iteration\_name                      (optioneel) naamgeving van de iteratie modellen;
- iteration\_start\_count              (optioneel) op welk nummer de nummering begint, standaard 0.

Deze kenmerken worden als attributen van de class bewaard en gelden voor alle iteraties die uitgevoerd worden vanuit deze class. Een iteratie wordt uitgevoerd met de functie **create\_iteration**. Bij create\_iteration hoeven slechts twee inputs gegeven te worden: de id's van de crosssection definitions die aangepast moeten worden en de kenmerken van het nieuwe trapezium profiel (bottom\_width, slope\_l, slope\_r, depth). Op dit moment is het nog niet mogelijk om met verschillende waarde voor slope\_l en slope\_r te rekenen. Create\_iteration maakt een nieuwe crosssection definition aan, waarbij het de gekozen crosssection definition ID's aanpast naar de nieuwe trapezium dimensies. De bodemhoogte blijft hierbij gelijk. Daarna wordt hier een nieuwe MDU bij gemaakt met het actuele iteratie nummer (deze wordt automatisch verhoogd iedere iteratie), deze MDU gebruikt alle bestanden van het bron-model samen met de nieuwe crosssection definition. Vervolgens wordt een DIMR aangemaakt voor deze MDU.

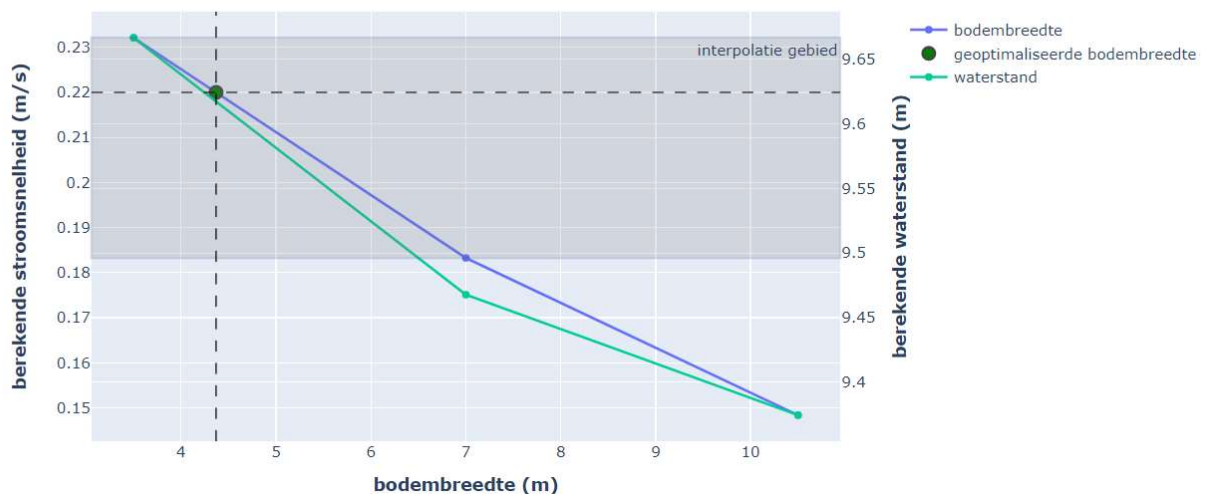
Vanuit de class kan het model ook doorgerekend worden met de functie **read\_latest**. Read\_latest kan op basis van het iteratie nummer in de class bepalen wat de nieuwste berekening is en rekent deze met de DIMR door. Er is ook een handmatige functie: **run\_model** waarbij de gebruiker zelf de locatie van het bat bestand en de model map moet verwijzen.

Deze iteraties staan allemaal in de work\_dir. Wanneer de gebruiker klaar is met itereren, kan het definitieve model geëxporteerd worden naar de output\_dir met de functie **export\_model**. Als de gebruiker geen input meegeeft aan deze functie, dan wordt het laatste model geëxporteerd naar de output map. Als de gebruiker een specific\_iteration opgeeft, dan wordt die iteratie geëxporteerd. Standaard wordt vervolgens de work map geleegd, tenzij de gebruiker cleanup=False heeft opgegeven.

Tot slot: **find\_optimum**. Deze functie analyseert meerdere iteraties binnen de zoekruimte en toetst deze aan de streefwaarde. In het geval van de profile optimizer 0.1.2 voor TKI4: deze functie zoekt naar welke berekening de beste stroomsnelheid heeft (streefwaarde). Het optimum wordt lineair geïnterpoleerd tussen de bodembreedte boven de streefwaarde en onder de streefwaarde. Ter interpretatie wordt ook de berekende waterdiepte geplot.

In onderstaande grafiek zijn ter illustratie drie iteraties uitgevoerd. Een bodembreedte van 3,5 m, 7 m en 11,5 meter. De doel stroomsnelheid is 0,22 m/s. Bij 3,5 m bodembreedte is de stroomsnelheid te hoog, maar bij 7 meter is deze te laag. Via lineaire interpolatie wordt de optimale bodembreedte bepaald: 4,3 m. Deze bodembreedte wordt nog een keer doorgerekend worden ter controle.

Relatie tussen bodembreedte, stroomsnelheid en waterlevel bij het te optimaliseren profiel



### 5.1.3 postprocessing.py

In postprocessing zit een class waarmee D-HYDRO Suite 1D2D resultaten worden uitgelezen (Results) en een functie voor het visualiseren van profielen.

De **Results** class draait automatisch functies om resultaten van de laatste tijdstap van de berekening om te zetten naar resultaten op mesh nodes (rekenpunten) of mesh edges (ruimte tussen twee rekenpunten). De mesh nodes bevatten waterdiepte en waterstand en worden in de Results class opgeslagen als *gdf\_points*. De mesh edges bevatten stroomsnelheid en debiet, en worden als *gdf\_lines* opgeslagen (al is de geometrie wel een punt). Vervolgens kan met de functie **result\_at\_xy** de resultaten nabij een xy locatie opgezocht worden. Er is gekozen voor een ruimtelijke aanpak met uitlezen op xy coördinaat omdat in D-HYDRO Suite 1D2D geen vaste namen voor rekenpunten zit. Daarnaast zijn de functies **export\_result\_gpkg** en **export\_result\_shp** beschikbaar, deze exporteren de *gdf\_lines* en *gdf\_points* naar een output map naar keuze. De exports zijn respectievelijk naar de bestandsformaten geopackage en shapefile.

**plot\_profiles** is een functie waarmee een profiel in twee verschillende modellen met elkaar wordt vergeleken. Deze functie wordt in het jupyter notebook gebruikt om weer te geven wat het oorspronkelijke profiel was en hoe het geoptimaliseerde profiel er uit ziet.



#### 5.1.4 preprocessing.py

In preprocessing.py staan een aantal scripts waarmee de eerste inschatting voor bodembreedte bepaald kan worden. Ook staat in dit script een hulp-functie voor het vaststellen van de zoekruimte.

De formule van Manning wordt in diverse functies gebruikt:

$$V = k * R_h^{2/3} * S^{1/2}$$

Waarbij:

V = stroomsnelheid [m/s]

k = k Manning (ruwheid) [s/m<sup>1/3</sup>]

R<sub>h</sub> = Hydraulische radius [m]

S = helling [-]

Daarnaast wordt gebruik gemaakt van de formule:

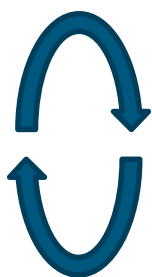
$$Q = v * A$$

Afvoer (Q) is gelijk aan de (gemiddelde) stroomsnelheid (v) maal het natte oppervlak (A).

**bottom\_width** is de eerste schatting voor bodembreedte. Het maakt gebruik van de formule van Manning en gebruikt als input een ruwheid (k Manning), helling (slope), talud, waterdiepte en een doel stroomsnelheid. De functie lost de formule van Manning op met de opgegeven inputs om een bodembreedte te schatten.

De initiële schatting van de bodemhoogte wordt nogmaals gecontroleerd voor de afvoer in de waterloop. Het oplossen van de formule van Manning in bottom\_width wordt namelijk niet opgelost voor een bepaald debiet, maar alleen voor een bepaald talud en waterdiepte voor een doel-stroomsnelheid. Het toetsen of het profiel bij het debiet past gebeurt in de functie **check\_QVA**. De functie begint met een initiële bodembreedte, profielkenmerken, helling, weerstand en een doel afvoer. Vervolgens worden de volgende stappen doorlopen:

1. Controleer of de bodembreedte groter is dan 0, zo niet, dan wordt de bodembreedte aangepast naar 1 meter breed.
2. Bereken de stroomsnelheid volgens Manning bij dit profiel.
3. Bereken hoeveel afvoer er met dit profiel en deze stroomsnelheid is.
4. Bepaal hoeveel de afvoer afwijkt van de gewenste afvoer.  
→ Zodra de afwijking kleiner is dan een gekozen toegestane waarde, is de check compleet en wordt de huidige bodembreedte gebruikt.
5. Als er een afwijking is dan wordt de bodembreedte met 5% aangepast (breder als er te weinig afvoer is, smaller als er teveel afvoer is.  
Hierna wordt stap 2 t/m 4 herhaald, gevolgd door stap 5 als de afwijking van het debiet nog steeds te groot is. Deze herhaling kan maximaal 30 keer plaatsvinden.



Met de functie **search\_window** kan vanaf een startwaarde een zoekruimte gedefinieerd worden aan de hand van een maximum marge en een maximaal aantal iteraties. Bijvoorbeeld bij een startwaarde van 4 meter breedte, 5 iteraties en een bandbreedte van 50% zou je de volgende zoekruimte krijgen: 2, 3, 4, 5, en 6 meter.