

## Functioneel Ontwerp hatyan getijanalyse in Python



## Functioneel Ontwerp hatyan getijanalyse in Python




<b>Opdrachtgever</b>	Rijkswaterstaat Water, Verkeer en Leefomgeving
<b>Contactpersoon</b>	Martin Scholten, Jan Rolf Hendriks
<b>Trefwoorden</b>	Functioneel Ontwerp, hatyan, getijanalyse, getijpredictie

### Documentgegevens

<b>Versie</b>	1.0
<b>Datum</b>	17-12-2020
<b>Projectnummer</b>	11205257-013
<b>Document ID</b>	11205257-013-DSC-0002
<b>Pagina's</b>	18
<b>Status</b>	definitief

### Auteurs(s)

	Jelmer Veenstra	
--	-----------------	--

Doc. Versie	Auteur	Controle	Akkoord	Publicatie
1.0	Jelmer Veenstra 	Jaco Stout 	Hans van Putten 	

# Inhoud

<b>1</b>	<b>Inleiding</b>	<b>4</b>
1.1	Doel van dit document	4
1.2	Documentstructuur	4
<b>2</b>	<b>Systeemoverzicht</b>	<b>5</b>
2.1	Totstandkoming	5
2.2	De systeemcontext en werkproces	6
<b>3</b>	<b>Eisen voor de applicatie</b>	<b>8</b>
3.1	Systeemeisen	8
3.2	Eisen voor interactie en werking van hatyan	8
3.3	Eisen voor inlezen van bestanden	9
3.4	Eisen voor achterliggende berekeningen	9
3.5	Eisen voor analyse, predictie en hoog/laagwaters	9
3.6	Eisen voor logging en reproduceerbaarheid	9
3.7	Afbakening	10
<b>4</b>	<b>Beschrijving van de functionaliteiten</b>	<b>11</b>
4.1	Inleiding	11
4.2	wrapper_RWS.py	11
4.3	components.py	12
4.4	timeseries.py	12
4.5	hatyan_core.py	15
4.6	analysis_prediction.py	17
<b>5</b>	<b>Referenties</b>	<b>18</b>

# 1 Inleiding

## 1.1 Doel van dit document

Dit document beschrijft het functionele ontwerp van de applicatie voor de voorspelling van astronomisch getij. Deze applicatie behoort tot de standaard software-suite waarmee RWS-CIV Afdeling Data Management Center haar dataservices richting gebruikers faciliteert. Het Functioneel Ontwerp (FO) wordt gebruikt als basis voor de verdere applicatieontwikkeling van hatyan, het geeft een conceptuele beschrijving van de structuur van de hatyan applicatie. Normaliter wordt een FO voorafgaand aan het ontwikkelen van een applicatie geschreven, in dit geval is dit proces omgedraaid. Ook nu is het FO belangrijk, omdat het nu en in de toekomst voorziet in een uitgebreid overzicht van welke functionaliteit er is gebouwd, bijvoorbeeld als de applicatie in de toekomst opnieuw in een nieuwe programmeertaal overgezet zou moeten worden.

## 1.2 Documentstructuur

Door dit alternatieve verloop van de planning zijn enkele onderwerpen die normaal in een FO staan al opgeschreven in de hatyan gebruikershandleiding (Veenstra en Kerkhoven, 2019), bijvoorbeeld:

- Welke workflows kan de gebruiker doorlopen
- Welke eenmalige handelingen kan of moet de gebruiker uitvoeren, zoals installatie en de eenmalige test

Dit FO bevat daarom alleen het volgende:

- Een systeemoverzicht (Hoofdstuk 2)
- De eisen voor de applicatie (Hoofdstuk 3)
- Een gedetailleerde beschrijving van de werking van de belangrijkste functies (Hoofdstuk 4)



## 2 Systeemoverzicht

### 2.1 Totstandkoming

Het voorspellen van astronomisch getijden speelt een belangrijke rol in diverse werkprocessen en producten van Rijkswaterstaat en daarbuiten. Het gaat enerzijds om de getijtafels van Nederland en diverse daaraan gerelateerde producten. Anderzijds speelt getijanalyse een belangrijke rol bij het maken van operationele waterstandsverwachtingen binnen de omgeving van het Water Management Centrum Nederland (WMCN) - specifiek de onderdelen: WMCN HMC en WMCN Kust en Benedenrivieren

De basis voor de methodiek van het geautomatiseerd voorspellen van astronomische getijden is gelegd in de periode 1965-1980. In die periode is ook de basis gelegd van het momenteel nog in gebruik zijnde Fortran-programma HATYAN. De onderliggende theoretische kennis van harmonische analyse is gebaseerd op Schureman (1940).

Het project “Verbetering Astronomisch getij” is een meerjarig KPP-project bij Deltares geweest in de periode 2015-2019, waarin is gewerkt aan het op peil brengen en houden van deze kennis over getijanalyse en -voorspelling en de modernisering van de bestaande HATYAN-software. Het project stelde als hoofddoel dat Rijkswaterstaat-Centrale Informatie Voorziening (RWS-CIV) met hatyan vanaf 2019 zelfstandig in de RWS-omgeving productie zou kunnen draaien voor het opstellen van het getijtafelboekje van 2020 en verder. Gelijktijdig is tijdens deze perioden bij een nieuwe generatie medewerkers van RWS en van Deltares het kennisniveau omtrent het voorspellen van astronomisch getij op peil gebracht en is toegewerkt naar een nieuwe versie van de programmatuur hatyan in Python.

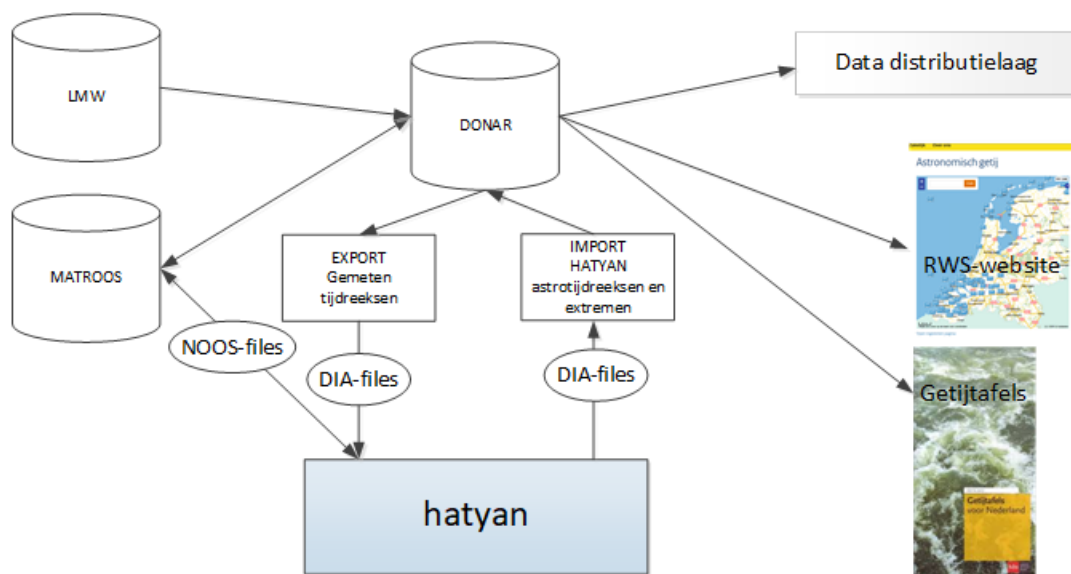
De applicatie hatyan is opgebouwd in Python versie 3.6. Het pakket wordt ingaande het jaar 2020 in opdracht van RWS als onderdeel van het KPP-programma door Deltares beheerd en onderhouden. De source-code, executables en de documentatie worden bij Deltares beheerd in Subversion (versiebeheer).

## 2.2 De systeemcontext en werkproces

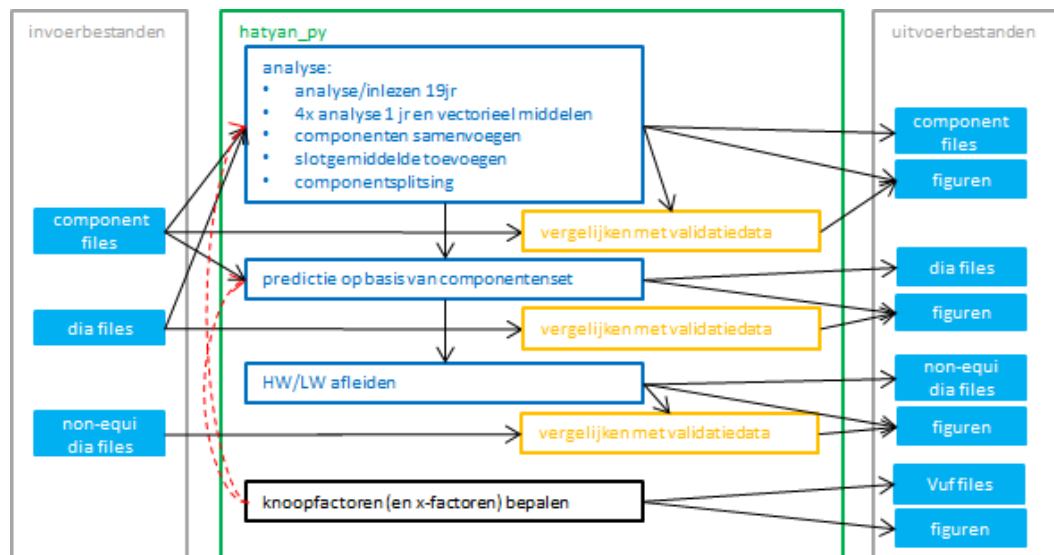
De applicatie hatyan is inpasbaar gemaakt binnen het huidige getijdeverwerkings proces, hetgeen betekent dat hatyan de gangbare invoerfiles kan lezen en de gewenste uitvoerfiles produceert en de oude Fortran-applicatie hiermee vervangen kan worden. Deze context is weergegeven in Figuur 2.1. Het blok hatyan is de applicatie zelf, waarvoor de interne stappen en de relaties met in- en uitvoerbestanden zijn uitgewerkt in Figuur 2.2. De onderlinge relaties tussen programma onderdelen zijn weergegeven in Figuur 2.3, alle scripts en functies zijn in meer detail beschreven in Hoofdstuk 4.

De applicatie hatyan is opgeleverd als een zelfstandig te gebruiken applicatie en zal binnen de ICT-omgeving van Rijkswaterstaat gaan draaien (zie Figuur 2.1). De applicatie gaat gebruikt worden door circa 5 medewerkers van RWS-CIV. Het voornaamste doel van de applicatie is om de gegevens, zoals deze tot op heden jaarlijks in de getijtafelboekjes worden gepubliceerd, te produceren. Het in Python geprogrammeerde hatyan dient daarmee dus een volwaardig alternatief te bieden voor de huidige op FORTRAN-code gebaseerde HATYAN-applicatie.

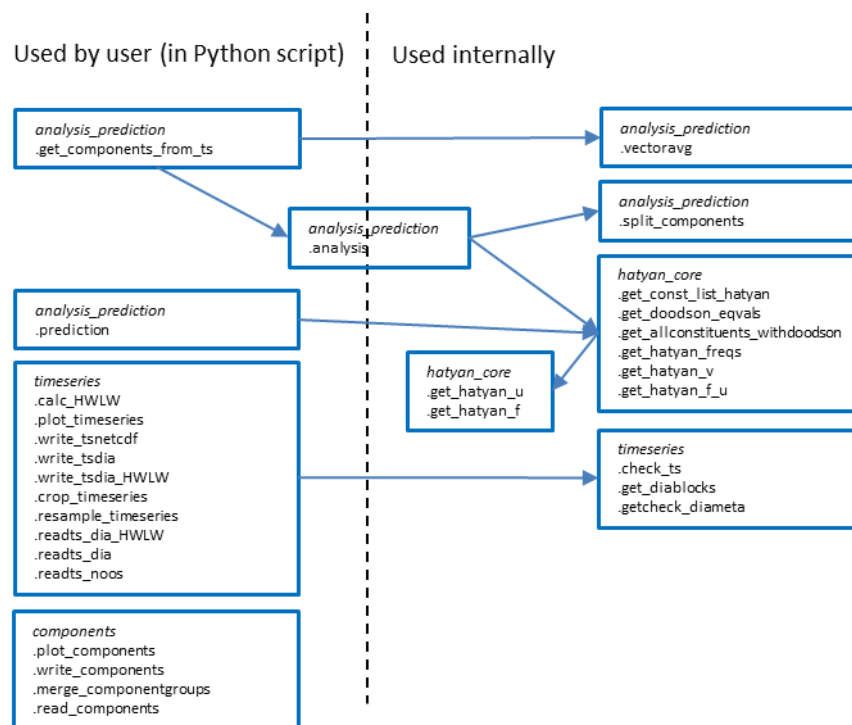
De programmatuur dient modulair van opzet te zijn en de berekeningen en datastructuren dienen aan te sluiten bij de huidige inzichten en technieken ten aanzien van wiskundige algoritmes, programmeerstructuren en hardware, teneinde een toekomstbestendig, gebruiksvriendelijk en onderhoudbaar softwarepakket op te leveren.



Figuur 2.1 De systeemcontext van hatyan



Figuur 2.2 Werkproces hatyan. Invoerbestanden (lichtblauwe blokken links) als tijdreeksen en analyseresultaten worden ingelezen en afhankelijk van de invoer in de configuratiefile worden de verschillende interne stappen doorlopen (alles binnen het groene blok), die vervolgens bijbehorende uitvoer (lichtblauwe blokken rechts) genereren. Het bepalen van knoopfactoren en x-factoren is een intern proces en speelt een rol bij zowel de analyse als de predictie. (Veenstra en Kerkhoven, 2019)



Figuur 2.3 De interne relaties van scripts (cursief) en functies (beginnend met punt). De richting van de pijlen geeft de aanroep aan, `analysis_prediction.prediction` roept bijvoorbeeld meerdere functies binnen `hatyan_core` aan.

## 3 Eisen voor de applicatie

Op basis van overleg met RWS zijn de globale eisen voor de applicatie opgesteld, deze zijn aangevuld met uiteindelijk ingebouwde functionaliteiten.

### 3.1 Systeemeisen

1. Platform: De routines/functies van hatyan worden opgezet in Python 3.6. De Python-applicatie zal gaan draaien op de RWS-Linux Servers (Redhat 6.8) – specifiek “de Getijserver”. Hiervoor dient een installatiehandleiding te worden geleverd.
2. Eigendom en licentie: Het intellectueel eigendom van hatyan en gerelateerde documenten ligt bij Rijkswaterstaat. Om verbeteringen aan de code te stimuleren wordt de broncode open source opgezet en komt onder GNU LGPL licentievoorwaarden beschikbaar voor derden.
3. Beheer: Functioneel beheer ligt bij RWS-WVL, Technisch beheer bij RWS-CIV. Deltares verzorgt het applicatiebeheer inclusief de daaronder aanwezige inhoud van FO (Functioneel Ontwerp), TO (Technisch Ontwerp), gebruikshandleiding (inclusief installatiehandleiding).
4. Versiebeheer: De code komt onder Subversion-versiebeheer bij Deltares vergelijkbaar met andere applicaties voor RWS. Hierbij komt ook releasemanagement.
5. Support: RWS wordt de primaire gebruiker van hatyan. Deltares zal wel faciliteren met een training aan gebruikers en een beslissingsboom.
6. Stabiliteit/consistentie: De code dient tenminste 5 jaar te kunnen functioneren, zonder dat grote inhoudelijke aanpassingen aan de code of methodiek moeten worden doorgevoerd.
7. Modulair: hatyan wordt modulair opgezet, zodat in een later stadium nieuwe functionaliteit en uitvoer-formats kunnen worden opgenomen. Bij het opzetten van de code wordt zo veel mogelijk rekening gehouden met in de toekomst gewenste flexibiliteit ten aanzien van bijvoorbeeld tijdsafhankelijk frequentie correctiefactoren en het toepassen van andere analyse en predictie methodieken.

### 3.2 Eisen voor interactie en werking van hatyan

1. De applicatie hatyan wordt via de commandline aangestuurd met een ‘configuratiefile’ als invoer. De configuratiefile is een python script dat ook op zichzelf gerund kan worden vanuit bijvoorbeeld Spyder. Hierin wordt hatyan aangeroepen en het bevat alle argumenten/instellingen, gegevens/bestandslocaties en bewerkingen die moeten worden uitgevoerd.
2. Omdat hatyan maar enkele keren per jaar wordt gebruikt, is de snelheid van de applicatie niet erg relevant. Wel moet deze zoveel mogelijk geoptimaliseerd worden door de Python code efficiënt te schrijven.



### 3.3 Eisen voor inlezen van bestanden

1. Het inlezen of wegschrijven van bestanden met tijdreeksen van waterstanden, te weten equidistante DIA-files (Donar Interface Asciifile) afkomstig uit DONAR en NOOS-bestanden afkomstig uit Matroos (deze laatste hoeven niet weggeschreven te worden). Bij beide bestanden is het van belang dat hiaatwaardes worden afgevangen en vervangen door NaN-waardes. Tot slot kunnen de bestanden een willekeurig tijdsinterval hebben, bijvoorbeeld 1, 10 of 60 minuten.
2. Het inlezen en wegschrijven van bestanden met hoog- en laagwaters, te weten non-equidistante dia-files afkomstig uit DONAR.
3. Het inlezen en wegschrijven van amplitudes en fases voor componenten, evenals middenstanden, uit of naar een analyseresultatenbestand.
4. Het wegschrijven van figuren met resultaten en validatievergelijkingen voor componenten, predicties en hoog/laagwaters
5. Zie de eisen voor analyse en predictie (Hoofdstuk 3.5) voor het praktische gebruik van deze invoer- en uitvoerbestanden

### 3.4 Eisen voor achterliggende berekeningen

1. Het bepalen van de Doodson getallen T, S, H, P, N en P1 voor een of meerdere opgegeven tijdstippen.
2. Het bepalen van frequenties en de waardes voor v op basis van doodson getallen, voor ieder van de 195 binnen hatyan beschikbare getijcomponenten. Deze getallen zijn nodig voor analyses en predicties.
3. Het bepalen van de waardes voor u en f (knoopfactoren) op basis van Doodson getallen, voor ieder van de 195 binnen hatyan beschikbare getijcomponenten. Deze getallen zijn nodig voor analyses en predicties.
4. Het uitvoeren van de v, u en f-waardes naar tekstbestanden

### 3.5 Eisen voor analyse, predictie en hoog/laagwaters

1. Een analyse uitvoeren voor een tijdreeks van verschillende lengtes, ingelezen uit een equidistante dia-file. In ieder geval het analyseren van:
  - a. een periode van enkele maanden tot 19 jaar in een keer
  - b. vier of minder losse kalenderjaren waarbij de resultaten vectorieel gemiddeld worden.
2. Het uitvoeren van componentensplitsing in geval van korte analyseperiodes
3. Opslaan van de analyseresultaten in een analyseresultatenbestand en in een figuur
4. Een predictie maken voor een bepaalde periode op basis middenstand, amplitudes en fases van een set van componenten, in hetzelfde hatyan proces berekend of ingelezen uit een eerder gegenereerd analyseresultatenbestand van hatyan of Fortran-HATYAN.
5. Opslaan van de predictie in een equidistante dia-file en in een figuur
6. Het berekenen van hoog- en laagwaters vanuit een predictie
7. Opslaan van de hoog- en laagwaters in een non-equidistante dia-file en in een figuur

### 3.6 Eisen voor logging en reproduceerbaarheid

1. Het uitvoeren van de doorlopen stappen in het hatyan proces in een logbestand
2. De uitvoer wordt in een lokale map opgeslagen, gekenmerkt met een timestamp van de hatyan run, met hierin ook een kopie van de gebruikte configuratiefile. Zo is altijd te herleiden op basis van welke datasets en welk type analyse/predictie de resulterende astronomische tijdreeksen en bijbehorende extremen tot stand zijn gekomen.

### 3.7 Afbakening

Geen onderdeel van de functionaliteit van hatyan vormen:

- Het uitvoeren van kwaliteitscontroles op de aangeboden waterstandsreeksen/analyses,
- Conversies/mutaties van aangeboden data formats (naast DIA en NOOS),
- Het controleren van de kwaliteit en compleetheid van de invoerdata
- Mutaties van aangeboden waterstandsreeksen, bijvoorbeeld het 'bijgissen' van data. De applicatie kan wel omgaan met hiaten in de data.

Verder geldt:

- Er worden in dit stadium nog geen inhoudelijke verbeteringen uitgevoerd aan de methodiek zoals deze is geïmplementeerd in de huidige versie van HATYAN. Door geen inhoudelijke wijzigingen door te voeren, wordt gegarandeerd dat in dit stadium de resultaten onderling vergelijkbaar en consistent blijven met de huidige oude HATYAN-versie.
- Er wordt geen grafische user-interface gemaakt. In plaats daarvan wordt gewerkt met command line en configuratiebestanden, zo wordt de reproduceerbaarheid gewaarborgd. Wel worden er figuren weergegeven als de gebruiker dit wenst. Deze figuren zijn interactief en de gebruiker kan bijvoorbeeld inzoomen en de plot onder een andere naam opslaan indien gewenst.
- Er wordt geen rekening gehouden met het maken van syntheses/predicties of analyses voor buitenlandse locaties anders dan enkele Engelse stations aan de Noordzeekust.

## 4 Beschrijving van de functionaliteiten

### 4.1 Inleiding

Deze paragraaf beschrijft de decompositie van de verschillende routines binnen hatyan. De volgende scripts met routines zijn gedefinieerd; elk ervan wordt in aparte secties besproken:

- wrapper\_RWS.py
- components.py
- timeseries.py
- hatyan\_core.py
- analysis\_prediction.py

Er zijn twee basisobjecten gedefinieerd welke binnen het hele hatyan proces worden gebruikt, beide zijn pandas DataFrames (een veelgebruikte Python class, (The pandas development team, 2020)) met enkele kolommen:

- pandas DataFrame met componentengegevens, kolommen:
  - o const\_list: de lijst van componenten
  - o A: de amplitudes voor ieder van de componenten
  - o phi\_rad: de fases van ieder van de componenten
  - o H0: de middenstand van de geanalyseerde tijdreeksen waaruit de componenten volgen
- pandas DataFrame met tijdreeksen, kolommen:
  - o times: tijden van de tijdreeks
  - o values: waarden van de tijdreeks
  - o HWLWcode: hoog- en laagwater codes in geval van een extremen tijdreeks

De functies die direct gerelateerd zijn aan de pandas DataFrames met componentengegevens en tijdreeksen staan in components.py en timeseries.py.

### 4.2 wrapper\_RWS.py

Een hulpscript met functies relevant voor het gebruik binnen RWS.

#### 4.2.1 init\_RWS()

initialiseert het hatyan proces. Dit is niet noodzakelijk, maar het zorgt ervoor dat er een header wordt geprint met alle relevante informatie over het huidige hatyan proces, wat dus ook terecht komt in de file met diagnostische uitvoer. Het bepaalt ook de starttijd van het hatyan proces, waarmee door exit\_RWS() de doorlooptijd bepaald kan worden. Tot slot wordt er een file `__NOT_FINISHED__` in de uitvoerfolder aangemaakt, welke na succesvolle afronding wordt verwijderd door exit\_RWS().

#### 4.2.2 exit\_RWS()

print de afrondingsinformatie van het hatyan proces, wat dus ook in de diagnostische file komt. Hierin staat onder andere de doorlooptijd van het proces.

#### 4.2.3 get\_outputfoldername()

Genereert de naam van de uitvoermap bij het aanroepen van hatyan. Dit script bestaat als aparte functie omdat het zowel buiten als binnen hatyan wordt aangeroepen:

- Buiten hatyan, om te controleren of de meegegeven configuratiefile bestaat en de uitvoerlocatie voor de logfile te bepalen voordat de applicatie daadwerkelijk draait
- Binnen hatyan, om de uitvoerlocatie voor alle hatyan uitvoer te bepalen

## 4.3 components.py

Dit script bevat alle functies die relevant zijn omtrent componenten binnen hatyan.

### 4.3.1 read\_components()

Deze functie leest een bestand in met gegevens van een hatyan getijanalyse (afkomstig van hatyan of het oude Fortran HATYAN).

Invoer:

- filename: pad naar een componentenfile

Uitvoer:

- pandas DataFrame met componentengegevens

### 4.3.2 merge\_componentgroups()

Dit onderdeel faciliteert het samenvoegen van verschillende componentensets, bijvoorbeeld het samenvoegen van analyseresultaten van een tijdreeks van 19 jaar met die van een tijdreeks van 4 jaar, waar bij beide andere componenten zijn geanalyseerd.

Invoer:

- twee pandas DataFrames met componentengegevens
- lijst met componenten die vanuit de tweede naar de eerste set worden overgedragen

Uitvoer:

- pandas DataFrame met componentengegevens

### 4.3.3 replace\_H0\_by\_slotgem()

Het vervangen van de berekende middenstand door een extern berekend slotgemiddelde. De functie selecteert het slotgemiddelde voor het opgegeven station en voegt dit toe aan het pandas DataFrame met componentengegevens.

Invoer:

- pandas DataFrame met componentengegevens
- pad naar bestand met slotgemiddelden per station
- stationsnaam

Uitvoer:

- pandas DataFrame met componentengegevens waarbij de parameter H0 (de middenstand) vervangen is door het ingelezen slotgemiddelde

### 4.3.4 plot\_components()

Het weergeven van de data uit het componentenobject in een figuur. Deze figuur kan aangepast worden en opgeslagen worden.

### 4.3.5 write\_components()

Wegschrijven van de data uit het pandas DataFrame met componentengegevens naar een bestand met componentengegevens, hierbij kan eventueel metadata meegegeven worden

## 4.4 timeseries.py

Dit script bevat alle functies die relevant zijn omtrent tijdreeksen binnen hatyan.

#### 4.4.1 **get\_diablocks()**

Hulpfunctie voor de functies `readts_dia` en `readts_dia_HWLW`. Deze functie haalt de verschillende blokken uit een DONAR-bestand (of dia-bestand) op en test of het opgevraagde station hierin beschikbaar is en of alle stations slechts eenmaal voorkomen.

Invoer:

- `filename`: bestandsnaam van het DONAR-bestand
- `station`: stationsnaam

Uitvoer:

- `block_starts`: de startregels van ieder blok
- `data_starts`: de startregels van de eigenlijke data binnen ieder blok
- `data_ends`: de laatste regels van de eigenlijke data binnen ieder blok
- `block_stations`: de stations corresponderend met ieder blok
- `block_id`: het nummer van ieder blok

#### 4.4.2 **getcheck\_diameta()**

Hulpfunctie voor de functies `readts_dia` en `readts_dia_HWLW`. Deze functie controleert of de metadata binnen het DONAR-bestand volgens verwachting is, bijvoorbeeld of de stationsnaam en het verticale referentievlak overeen komen met wat opgevraagd is.

Invoer:

- `data_meta`: metadata uit de diafile
- `station`: stationsnaam
- `diatype`: equidistant (tijdreeksen met waterstanden) of non-equidistant (tijdreeksen van hoog en laagwaters)

Uitvoer:

- `file_station_name`: stationsnaam in de metadata
- `file_vertref`: verticaal referentievlak in de metadata
- `times_fromfile`: de tijdreeksen die volgt uit de begintijd, eindtijd en tijdsinterval uit de metadata. Indien het een non-equidistante dia-file betreft
- `file_station_coord`: het coördinaat van het station

#### 4.4.3 **readts\_dia()**

Voor het inlezen van equidistante dia-files (extensies `*.dia` en `*.txt`). Hierbij worden de hulpfuncties `get_diablocks()` en `getcheck_diameta()` gebruikt. Hiaatwaardes (99) worden vervangen door nan-waardes.

Invoer:

- `pad` naar equidistante dia-file
- `stationsnaam`

Uitvoer:

- pandas DataFrame met tijdreeksen

#### 4.4.4 **readts\_dia\_HWLW()**

Deze functie leest een non-equidistante dia-file in, dit bestandstype bevat doorgaans tijdstippen en waardes voor hoog- en laagwaters. Hierbij worden de hulpfuncties `get_diablocks` en `getcheck_diameta` gebruikt. Hiaatwaardes (99) worden vervangen door nan-waardes.

Invoer:

- `pad` naar niet-equidistante dia-file
- `stationsnaam`

Uitvoer:



- pandas DataFrame met tijdreeks, inclusief kolom met hoog- en laagwater codes

#### 4.4.5 **readts\_noos()**

Voor het inlezen van matroosbestanden (extensie \*.noos).

Invoer:

- pad naar noos file

Uitvoer:

- pandas DataFrame met tijdreeks

#### 4.4.6 **check\_ts()**

Print gegevens over de tijdreeks, wordt gebruikt in alle functies die tijdreeksen inlezen of bewerken, zodat de diagnostische file duidelijk het verloop van de data laat zien.

Invoer:

- pandas DataFrame met tijdreeks

Uitvoer

- prints van tijdstap in minuten, enkele tijdstappen, lengte, percentage en aantal nan-waardes.

#### 4.4.7 **resample\_timeseries()**

Het opnieuw definiëren van de tijdreeks met de opgegeven tijdstap

Invoer:

- pandas DataFrame met tijdreeks
- tijdstap in minuten

Uitvoer:

- pandas DataFrame met tijdreeks (resampled)

#### 4.4.8 **crop\_timeseries()**

Het bijsnijden van een deel van de tijdreeks op basis van een opgegeven begin- en eindtijd, waarbij gecontroleerd wordt of alle tussenliggende data beschikbaar is.

Invoer:

- pandas DataFrame met tijdreeks
- times\_ext: gewenste begin- en eindtijd

Uitvoer:

- pandas DataFrame met tijdreeks (bijgesneden)

#### 4.4.9 **calc\_HWLW()**

Het berekenen van de hoog- en laagwaters van de opgegeven tijdreeks met een getijpredictie.

Invoer:

- pandas DataFrame met tijdreeks

Uitvoer:

- pandas DataFrame met tijdreeks van hoog- en laagwaters, inclusief de bijbehorende code

#### 4.4.10 **plot\_timeseries()**

het plotten van de tijdreeks met een meting of predictie, eventueel samen met:

- een tijdreeks uit validatiedata

- berekende hoog- en laagwaters
- hoog- en laagwaters van de validatiedata

Invoer:

- pandas DataFrame met tijdreeks
- optioneel: pandas DataFrames met tijdreeks met validatiedata en extremen

Uitvoer:

- fig,ax: het figuur en de subplots zodat de deze aangepast en opgeslagen kunnen worden.

#### 4.4.11 **write\_tsdia()**

het wegschrijven van een equidistante tijdreeks naar DONAR-formaat (dit wordt gebruikt voor predicties van waterstanden).

Invoer:

- pandas DataFrame met tijdreeks
- station: de stationsnaam
- vertref: het verticale referentievlak
- filename: de filename van het uitvoerbestand

Uitvoer:

- diabestand met gegevens van de equidistante tijdreeks

#### 4.4.12 **write\_tsdia\_HWLW()**

Het wegschrijven van een non-equidistante tijdreeks naar DONAR-formaat (dit wordt gebruikt voor hoog-en laagwaters).

Invoer:

- pandas DataFrame met niet tijdreeks van extremen
- station: de stationsnaam
- vertref: het verticale referentievlak
- filename: de filename van het uitvoerbestand

Uitvoer:

- diabestand met gegevens van de niet-equidistante tijdreeks

#### 4.4.13 **write\_tsnetcdf()**

Het wegschrijven van een netCDF file van de tijdreeks.

Invoer:

- ts: pandas DataFrame met tijdreeks
- station: de stationsnaam
- vertref: het verticale referentievlak
- ts\_ext: pandas DataFrame met tijdreeks van extremen (optioneel)
- tzone\_hr: de tijdzone, 1 in geval van MET (default waarde)

Uitvoer:

- netCDF met gegevens van de equidistante tijdreeks, deze file bevat de data voor alle stations en indien opgegeven ook de tijdreeksen met extremen (met heel veel nan-waardes)

## 4.5 **hatyan\_core.py**

Dit script bevat alle functies gerelateerd aan de getijcomponenten van hatyan. Het is rechtstreeks overgenomen uit het Fortran-HATYAN en bevat de formules waarmee de standen van verschillende hemellichamen berekend kunnen worden, evenals de frequenties, v, u en f-waardes voor alle 195 componenten beschikbaar binnen hatyan.

#### 4.5.1 **get\_const\_list\_hatyan()**

lijsten van componenten, namelijk alle 195, de standaardset van 94 (voor vierjarige analyse) en de lijst voor 19-jarige analyse (SA en SM).

#### 4.5.2 **get\_doodson\_eqvals()**

Deze functie berekent de waardes voor Doodsongetallen T, S, H, P, N en P1 voor het opgegeven tijdstip.

Invoer: datum waarvoor de getallen moeten worden berekend. De functie wordt aangeroepen vanuit verschillende functies binnen hatyan\_core.py

De Doodsongetallen zijn de volgende:

- Tj: het aantal Juliaanse eeuwen (=36525 dagen) die zijn verstreken sinds 31 december 1899 12.00 uur GMT, deze waarde wordt gebruikt als invoer voor S, H, P, N en P1.
- T: het verstreken aantal uren sinds middernacht of het middaguur in radialen, dit is een indicatie voor de hoeveelheid rotatie van de aarde om zijn eigen as ten opzichte van de zon.
- S: middelbare lengte van de maan
- H: middelbare lengte van de zon
- P: middelbare lengte van het perieum van de maan
- N: afstand van de klimmende knoop van de maan tot het lentepunt
- P1 of P': middelbare lengte van het perieum van de zon

#### 4.5.3 **get\_hat\_constants()**

functie waarmee drie constanten berekend worden die als invoer voor de functies voor u en f-waardes (knoopfactoren) nodig zijn. Deze functie haalt ook Doodson op met de functie get\_doodson\_eqvals.

#### 4.5.4 **get\_hatyan\_freqs()**

Invoer:

- const\_list: lijst met componenten

Uitvoer: lijst met frequenties voor ieder van de opgevraagde componenten.

#### 4.5.5 **get\_hatyan\_v()**

Invoer:

- const\_list: lijst met componenten
- dood\_date: datum waarvoor de getallen moeten worden berekend

Deze functie haalt ook Doodson op met de functie get\_doodson\_eqvals.

Uitvoer: lijst met v-waardes voor ieder van de opgevraagde componenten.

#### 4.5.6 **get\_hatyan\_u()**

Invoer:

- const\_list: lijst met componenten
- dood\_date: datum waarvoor de getallen moeten worden berekend

Deze functie haalt ook Doodson op met de functie get\_doodson\_eqvals.

Uitvoer: lijst met u-waardes voor ieder van de opgevraagde componenten.

#### 4.5.7 **get\_hatyan\_f()**

Invoer:

- const\_list: lijst met componenten
- dood\_date: datum waarvoor de getallen moeten worden berekend

Deze functie haalt ook Doodson op met de functie get\_doodson\_eqvals.

Uitvoer: lijst met f-waardes voor ieder van de opgevraagde componenten.

## 4.6 analysis\_prediction.py

Dit script bevat de eigenlijke functies voor getijanalyse en predictie, welke inhoudelijk zijn beschreven in het Technisch Ontwerp.

### 4.6.1 get\_components\_from\_ts()

Dit is een paraplufunctie voor getijanalyse. Bij gebruik van de standaard RWS methode voor analyse worden vier jaar achtereenvolgens geanalyseerd en vervolgens worden de resultaten voor amplitude en fases vectorieel gemiddeld. Dat is waar deze functie voor dient en ondertussen roept het de functie analysis (Hoofdstuk 4.6.2) meerdere malen aan.

Invoer:

- ts: pandas DataFrame met tijdreeks
- const\_list: lijst met te analyseren componenten
- instellingen voor knoopfactoren, xfactor etc
- gegevens voor componentensplitsing (optioneel)

Uitvoer:

- pandas DataFrame met componentengegevens
- optioneel: pandas DataFrame met componentengegevens voor afzonderlijk geanalyseerde jaren

### 4.6.2 analysis()

De eigenlijke analysefunctie waarbij de een ingelezen tijdserie wordt omgezet in amplituden en fasen voor alle opgevraagde componenten. Ook kan hierbij componentensplitsing worden toegepast. Binnen deze functie worden de frequenties en waarden voor v, u en f opgevraagd uit de relevante functies in hatyan\_core.py.

Invoer:

- gelijk aan get\_components\_from\_ts()

Uitvoer:

- gelijk aan get\_components\_from\_ts()

### 4.6.3 prediction()

De eigenlijke functie voor het maken van een getijpredictie. Binnen deze functie worden de frequenties en waarden voor v, u en f opgevraagd uit de relevante functies in hatyan\_core.py.

Invoer:

- comp: pandas DataFrame met componentengegevens
- times\_pred\_all: de tijdreeks waarvoor de predictie gemaakt moet worden, of:
  - o times\_ext: de extent van de te maken predictie (start en stop datum/tijd)
  - o timestep\_min: de tijdstap van de te maken precitie (in minuten)
- instellingen als knoopfactoren en X-facotr

Uitvoer:

- pandas DataFrame met tijdreeks

## 5 Referenties

Irazoqui Apecechea, M. (2018), *HATYAN technical design*. Deltares rapport 11202223-000-ZKS-0002, 27 Februari 2018 (gerefereerd als Technisch Ontwerp)

Schureman, P. (1940). *The manual of Harmonic Analysis and Prediction of Tides, revised edition*. U.S. Coast and Geodetic Survey

The Pandas development team (2020), *pandas.DataFrame*. weblink:  
<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>

Veenstra, J.; Kerkhoven, D. (2019). *Gebruikershandleiding hatyan getijanalyse*. Deltares rapport 11203717-000-ZKS-0008 versie 1.0, 3 december 2019 (concept)