

GETIJSPECIALS ONDER MELISSA

Documentatie

27 oktober 2000

in opdracht van RIKZ

GETIJSPECIALS ONDER MELISSA
Documentatie

27 oktober 2000

i.o.v. RIKZ

MobiData
Rotterdamse Rijnweg 126
3042 AS Rotterdam

tel. 010-2621422

Auteur: N.J. van der Zijpp

Inhoud

1	Installatie.....	1
2	Algemeen.....	2
3	Onderlinge samenhang – aanroep structuur routines.....	3
3.1	Overzicht Getijspecial-entrypoints.....	3
3.2	Structuur van de Getijspecial-programmatuur.....	4
4	Documentatie van Matlab source code.....	8
5	HTML Documentatie.....	8
	Appendix A: Source code documentatie.....	9

Voorwoord

Voor u ligt het rapport 'Getijspecials onder Melissa, Documentatie'. Dit rapport is één van de producten van het project 'Getijspecials' dat door MobiData is uitgevoerd in opdracht van RIKZ. De Getijspecial-functies zijn in te delen in twee categorieën:

- Getijspecials onder Melissa
- Getij specials in C

In beide gevallen gaat het om dezelfde functies, echter de getijspecials onder Melissa zijn aan te sturen uit de Melissa interface en met behulp van Melissa Macro opdrachten, terwijl de Getijspecials in C in- en uitvoer plegen via DONAR Interface Files (in ASCII formaat: DIA). De implementatie van deze twee categorieën van functies is ondergebracht in twee deelprojecten. Het eerste deelproject (Getijspecials onder Melissa) is afgerond in 1999. Het tweede deelproject is afgerond in 2000.

Deze rapportage is een herziene versie van het rapport 'Getijspecials, Documentatie' (MobiData i.o.v. RIKZ, December 1999). Door deze herziening sluit de rapportage ook aan bij het project 'Getijspecials in C'.

De producten die in het kader van de twee deelprojecten zijn opgeleverd zijn de volgende:

- Het rapport, 'Getijspecials, Technische Specificatie'
- Het onderhavige rapport 'Getijspecials onder Melissa, Documentatie'
- De Matlab source code van de Getijspecials onder Melissa op CD ROM
- De HTML documentatie van de Getijspecials onder Melissa op CD ROM
- Het rapport 'Getijspecials in C, Documentatie'
- De Matlab en C source code van de Getijspecials in C op CD ROM
- De .dll en .exe files t.b.v. de Getijspecials in C op CD ROM
- De HTML documentatie van de Getijspecials in C op CD ROM

1 Installatie

De getijspecial software bestaat uit de volgende files:

```
aardappels.m  
cubspline.m  
dispperiode.m  
dutchmonth.m  
f_agger.m  
f_eqfit.m  
f_havenget.m  
f_hooglaag.m  
f_laaglaag.m  
f_ongelijkheid.m  
f_tussenschr.m  
f_vliegplan.m  
f_zuiver.m  
filterandsmooth.m  
getij.m  
getstatuscodes.m  
group.m  
indexmax.m  
mls_agger.m  
mls_eqfit.m  
mls_expgdps.m  
mls_export.m  
mls_havenget.m  
mls_hooglaag.m  
mls_laaglaag.m  
mls_ongelijkheid.m  
mls_tussenschr.m  
mls_vliegplan.m  
mls_zuiver.m  
pix2norm.m  
synchronize.m  
tijd2zomertijd.m  
vliegopt.m
```

Deze files werken samen met de Melissa software. Het is aan te bevelen de files in een aparte directory te plaatsen. Bijvoorbeeld de directory (relatief tot de huidige Matlab directory):

```
getijspecials
```

De installatie is compleet nadat de directory waarin de getijspecial files zijn geplaatst aan het Matlabpad is toegevoegd met het commando:

```
>> addpath getijspecials
```

De getijspecialsoftware wordt geactiveerd met het commando:

```
>> getij
```

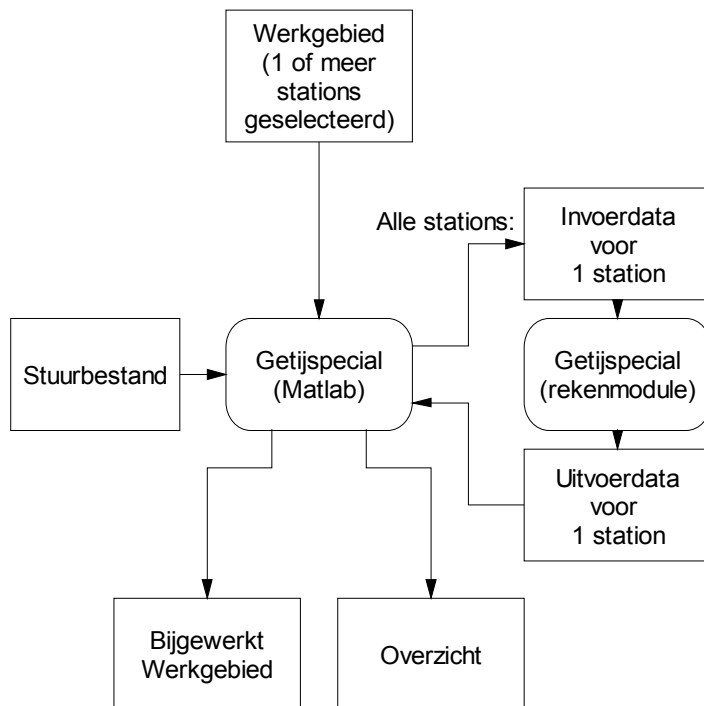
Dit commando start Melissa op en laadt de getijspecial menu's

2 Algemeen

Binnen Matlab bestaat de Getijspecial steeds uit twee functies:

- *De interface functie.* Deze functie leest en schrijft data naar een werkgebied en interpreteert een stuurfile. Deze functie schrijft ook een overzicht weg. Deze module fungeert dus als ‘schil’ om de rekenmodule.
- *De feitelijke rekenmodule.* Deze maakt voor één station tegelijk een berekening. Deze functie wordt herhaaldelijk aangeroepen wanneer er meerdere stations zijn geselecteerd. De bijbehorende filenaam is steeds f_XXXX, met XXXX de naam van de aanroepende interface functie

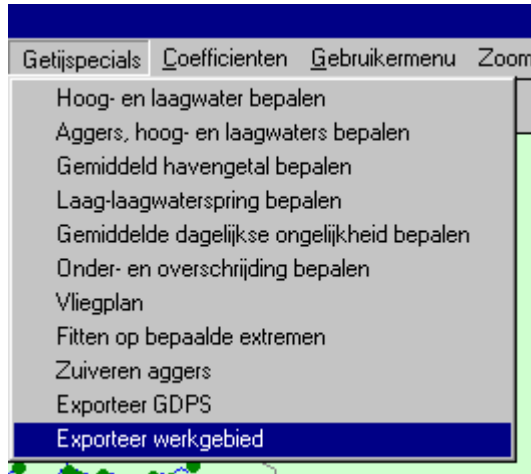
De rekenmodule is in Matlab geschreven maar is zo opgezet dat het makkelijk is om deze functie in C te implementeren.



3 Onderlinge samenhang – aanroep structuur routines

3.1 Overzicht Getijspecial-entypoints

Het intikken van het commando ‘getij’ heeft tot gevolg dat het menu ‘Getijspecials’ in de menubalk van het Melissa hoofdscherm wordt ingevoegd.



Figuur 1: Vanuit menu aan te roepen Getijspecial functies

De onderstaande tabel geeft per menu optie het zogenaamde ‘entrypoint’ in de getijspecial Matlab programmatuur.

Menuoptie	Matlab m-file entrypoint
Hoog- en laagwater bepalen	mls hooglaag
Aggers, hoog- en laagwaters bepalen	mls agger
Gemiddeld havengetal bepalen	mls havenget
Laag-laagwaterspring bepalen	mls laaglaag
Gemiddelde dagelijkse ongelijkheid bepalen	mls ongelijkheid
Onder- en overschrijding bepalen	mls tussenschr
Vliegplan	mls vliegplan
Fitten op bepaalde extremen	mls eqfit
Zuiveren aggers	mls zuiver
Exporteer GDPS	mls expgdps
Exporteer werkgebied	mls export

3.2 Structuur van de Getijspecial-programmatuur

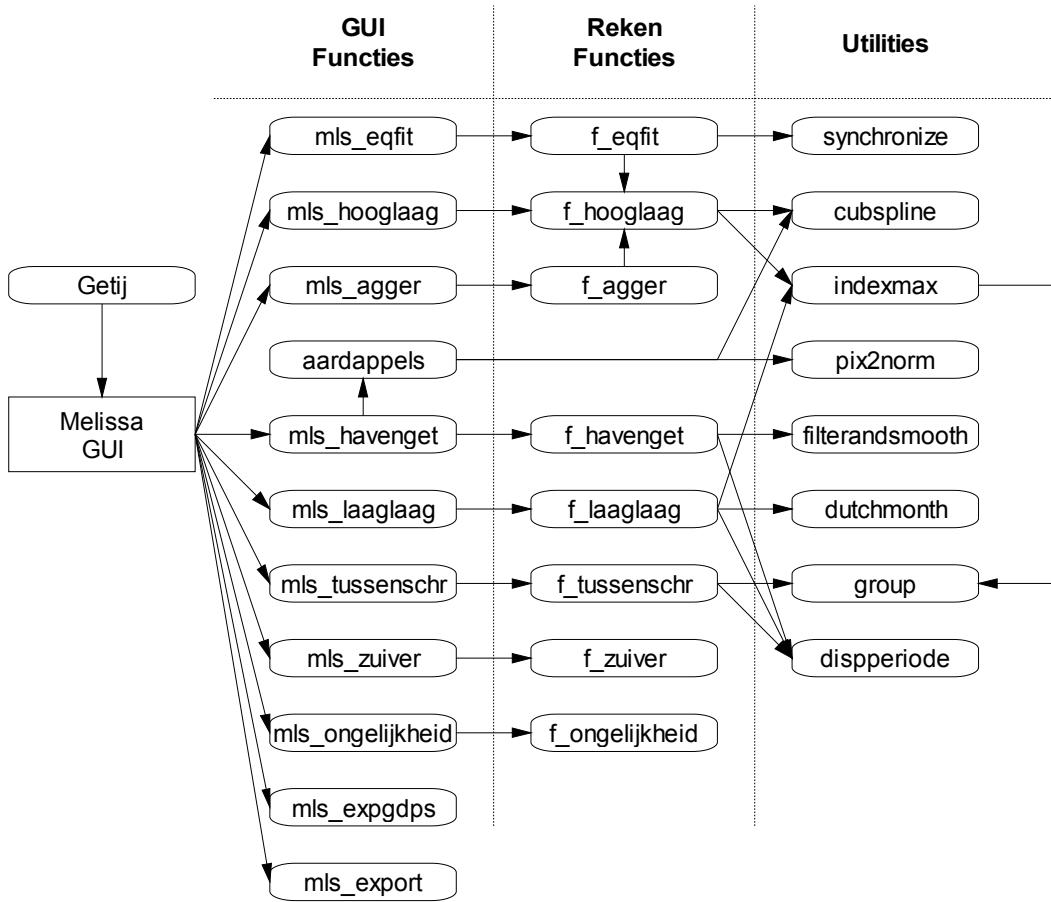
De Getijspecial matlab functies zijn in vier categorieën in te delen:

- *GUI functies*. Deze functies zorgen voor de grafische interface. Ze bevatten aanroepen naar z.g. Matlab Handle Graphics objects, zoals figures, axes, uimenu's en uicontrols. Zie de Matlab Manual voor meer informatie.
- *Rekenfuncties*. Dit zijn functies die een berekening uitvoeren. Er komen in de rekenfuncties geen grafische bewerkingen voor (dus geen aanroepen naar Handle Graphics objecten). De rekenmodules hebben steeds op slechts één station tegelijk betrekking. Binnen de Getijspecials zijn de reken modules te herkennen aan het voorvoegsel 'f_'.
- *Utilities*. Dit zijn af te zonderen elementen van rekenmodules die op verschillende plaatsen bruikbaar zijn en daarom in een aparte functie zijn ondergebracht.

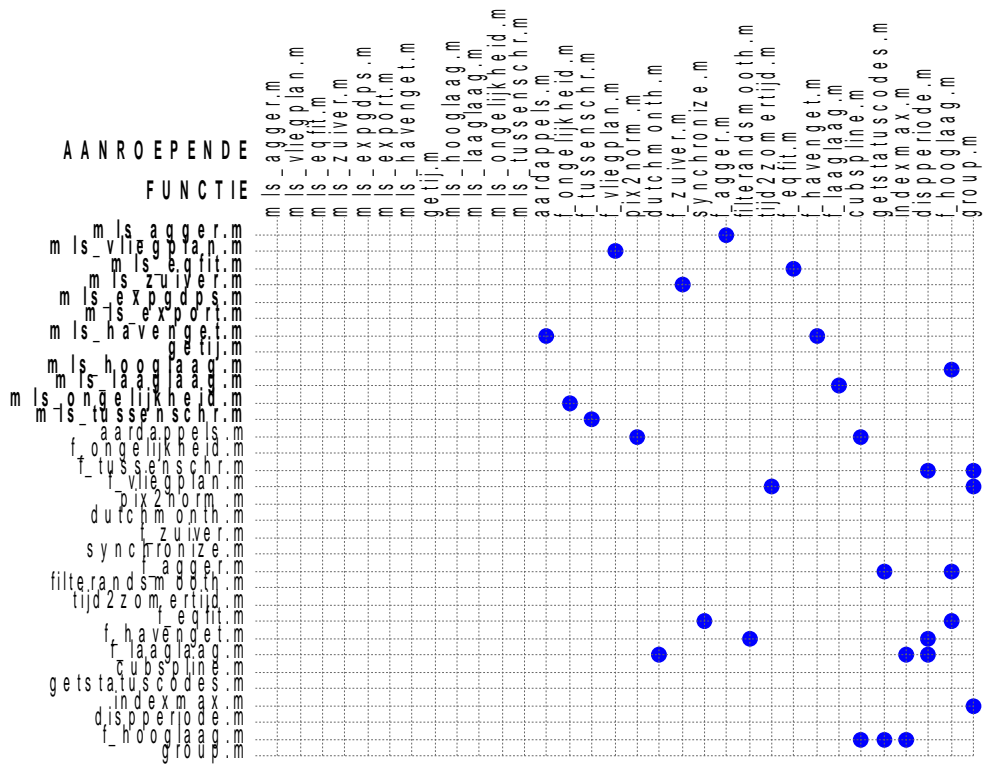
De onderstaande tabel geeft een overzicht van de getijspecial-functies en de categorie waartoe ze behoren.

GUI functies	Rekenmodules	Reken utilities
aardappels.m		
mls_agger.m	f_agger.m	
mls_eqfit.m	f_eqfit.m	
mls_export.m		
mls_expgdps		
mls_getij.m		
mls_havenget.m	f_havenget.m	
mls_hooglaag.m	f_hooglaag.m	
mls_laaglaag.m	f_laaglaag.m	
mls_ongelijkheid.m	f_ongelijkheid.m	
mls_tussenschr.m	f_tussenschr.m	
mls_zuiver.m	f_zuiver.m	
		cubspline.m dispperiode.m dutchmonth.m filterandsmooth.m group.m indexmax.m pix2norm.m synchronize.m

Figuur 2 laat zien op welke wijze de modules elkaar aanroepen. Figuur 2 is afgeleid uit Figuur 3.



Figuur 2: *Overzicht onderlinge aanroepen Getijspecials. Elke pijl staat voor een functieaanroep; de functie waar de pijl naar wijst wordt aangeroepen door de functie waar de pijl vandaan komt.*



Figuur 3: Afhangelijkheidsdiagram: een stip in kolom j van rij i betekent dat routine j wordt aangeroepen door routine i

Tenslotte laat de onderstaande lijst zien - per functie - welke modules elkaar aanroepen. Dit is gedaan door steeds de aanroepende routine bovenaan te zetten. De routines die worden aangeroepen staan eronder, één positie ingesprongen.

Activeren Getijspecials

getij

Harbek10: Bepalen hoog- en laagwater

mls_hooglaag

- f_hooglaag
- cubspline
- indexmax

Jaboek25: Bepaling hoog- en laagwater en aggers

mls_agger

- f_agger
- f_hooglaag
- cubspline
- indexmax

gtyber31: Bepalen gemiddeld havengetal

mls_havenget

- f_havenget
- filterandsmooth
- disperiode
- aardappels
- cubspline
- pix2norm

gtyber50: Bepalen laag-laagwaterspring

mls_laaglaag
 f_laaglaag
 disperiode
 indexmax
 dutchmonth

gtyber12: Bepalen gemiddelde dagelijkse ongelijkheid

mls_ongelijkheid
 f_ongelijkheid

jaboek60: Bepalen over-, onder en tussenschrijding

mls_tussenschr
 f_tussenschr
 disperiode
 group

htycur10: Fitten van equidistante reeks op bepaalde extremen

mls_eqfit
 f_eqfit
 f_hooglaag
 cubspline
 indexmax
 synchronize

jaboek16: Zuiveren van aggers

mls_zuiver
 f_zuiver

Exporteren

mls_export

4 Documentatie van Matlab source code

Appendix A bevat van alle routines een beknopte beschrijving. Deze beschrijving omvat:

- INPUT: invoervariabelen van de routine
- OUTPUT: uitvoervariabelen van de routine
- INGELEZEN FILES: files waaruit gelezen wordt (indien van toepassing)
- WEGGESCHREVEN FILES: files waarnaar geschreven wordt (indien van toepassing)
- USER INPUT: mogelijkheden voor gebruikersinvoer van toetsenbord of muis (indien van toepassing)
- OUTPUT NAAR SCHERM: gegevens die naar het scherm worden geschreven (indien van toepassing)
- WERKWIJZE: een beschrijving van de werking van de routine

De routines worden genoemd in alfabetische volgorde.

Dezelfde beschrijving is terug te vinden in de header van de files zelf.

5 HTML Documentatie

Er is ook een HTML documentatie gegenereerd. Deze is samen met de sources geleverd op een CD-rom. Per module is een HTML document gegenereerd met de volgende informatie:

- Naam en locatie van de file
- Aanroep formaat
- Functie omschrijving (Deze is identiek aan de beschrijving in hoofdstuk 2)
- Lijst van aangeroepen modules vanuit deze module
- Lijst van modules die deze module aanroepen

De HTML tekst is voorzien van 'Hyperlinks' waarmee snel van de ene naar de andere module kan worden gesprongen.

Er is een apart HTML document met de naam 'index.htm'. Dit document bevat een index van alle functies.

Appendix A: Source code documentatie

```

=====
DIRECTORY: d:\d\melissa\develop\getijspecials
=====

INHOUD:
aardappels.m
cdg.m
cubspline.m
dispperiode.m
dutchmonth.m
f_agger.m
f_eqfit.m
f_havenget.m
f_hooglaag.m
f_laaglaag.m
f_ongelijkheid.m
f_tussenschr.m
f_vliegplan.m
f_zuiver.m
filterandsmooth.m
getij.m
getstatuscodes.m
group.m
indexmax.m
mls_agger.m
mls_eqfit.m
mls_expgdps.m
mls_export.m
mls_havenget.m
mls_hooglaag.m
mls_laaglaag.m
mls_ongelijkheid.m
mls_tussenschr.m
mls_vliegplan.m
mls_zuiver.m
pix2norm.m
synchronize.m
tijd2zomertijd.m

*****
FILE: aardappels.m

    aardappels - plot aardappelgrafiek

CALL
    aardappels(optie,data);

INPUT
    optie
    data: structure array met de volgende velden
        loccode      : stationsaanduiding
        TculmHW      : culminatietijdstip (ruwe data) HW
        StandHW      : voortschreidend gemiddelde/mediaan stand (ruwe data)HW
        HavengetHW   : voortschreidend gemiddelde/mediaan havengetal (ruwe data)HW
        TculmLW      : idem LW
        StandLW      : idem LW
        HavengetLW   : idem LW
        StandHWint   : voortschreidend gemiddelde/mediaan stand
                        (geinterpoleerde data)HW
        HavengetHWint: voortschreidend gemiddelde/mediaan havengetal
                        (geinterpoleerde data)HW
        StandLWint   : idem LW
        HavengetLWint: idem LW

OPMERKING: StandHWint, HavengetHWint, StandLWint, HavengetLWint hebben betrekking
           op de interpolatietijdstippen TGRID (momenteel ingesteld op hele minuten)

OUTPUT

USER INPUT
    Gebruiker kan stations selecteren,

```

HW of LW selecteren,
Menus kiezen, Filenamen ingeven, etc.

OUTPUT NAAR SCHERM
Diverse plots

WERKWIJZE

Het gedrag van deze functie hangt af van de invoer parameter 'optie'

switch optie

case 'display'
bepaal het geselecteerde station

Als get menu HW is geselecteerd, plot dan de aardappelgrafiek voor hoogwater
Dit gebeurt in de subfunctie 'showdata'

Als get menu LW is geselecteerd, plot dan de aardappelgrafiek voor laagwater
Dit gebeurt in de subfunctie 'showdata'

Display de naam van het station en de duur van de analyseperiode

Format en display de gewichten voor stand en havengetal die
tijdens interpolatie worden gebruikt

Als get menu HW is geselecteerd, plot dan de markeringslijnen en text
voor hoogwater
Dit gebeurt in de subfunctie 'showlines'
Dit gebeurt pas hier omdat nu de xlim en ylim property van het figuur
niet meer veranderen

Voorzie de diverse assen van een tijdaanduiding in uren en minuten
Hiervoor wordt de utility tm_ax aangeroepen.

case 'updatePstand'

Deze aanroep volgt op het wijzigen van de gewichtsparameter voor
de interpolatie van de waterstand

Haal het current station op

Interpoleer de stand met opgegeven gewicht

De interpolatie gebeurt in de subfunctie interp_stand

Nadat de gegevens zijn bijgewerkt wordt alles opnieuw geplot

Dit gebeurt door de aanroep:

aardappels('display');

case 'updatePtijd'

Deze aanroep volgt op het wijzigen van de gewichtsparameter voor
de interpolatie van het havengetal

Haal het current station op

Interpoleer havengetal gegevens met opgegeven gewicht

De interpolatie gebeurt in de subfunctie interp_haven

Nadat de gegevens zijn bijgewerkt wordt alles opnieuw geplot

Dit gebeurt door de aanroep:

aardappels('display');

case 'load'

gebruiker voert filenaam in en data wordt geladen

case 'save'

gebruiker voert filenaam in en data wordt bewaard

case 'HW'

button HW ingedrukt ==> toggle HW button

Nadat menu is gewijzigd wordt alles opnieuw geplot

Dit gebeurt door de aanroep:

aardappels('display');

case 'LW'

button LW ingedrukt ==> toggle LW button

Nadat menu is gewijzigd wordt alles opnieuw geplot

Dit gebeurt door de aanroep:

aardappels('display');

case 'plotfrommenu'

De gebruiker heeft een stationnaam van een dropdown menu geselecteerd

Pas menu aan:

- current station is checked en disabled

- andere menus zijn unchecked en enabled

Nadat menu is gewijzigd wordt alles opnieuw geplot

Dit gebeurt door de aanroep:

aardappels('display');

Na het opnieuw displayen view maximaliseren en zoom history initialiseren

case 'install'

Verwijder figure met tag 'AARDAPPELWINTAG' (als het aanwezig is)

Maak een nieuw figure window aan voor de aardappelplot

Creeer de assen voor plotten Tijd, Stand en Aardappelgrafiek

```

Creeer de labels
Creeer de menus
Creeer de toolbar
zet alles op zijn plaats met:
  aardappels('resize');
  aardappels('display');
activeer zoomtool
case 'resize'
  Zet alle buttons uit de doolbar op zijn plaats
  Bepaal de positie van de axes, afhankelijk van aantal assen dat zichtbaar is
  Hierbij wordt de utility 'det_pos' gebruikt
  Bepaal de positie van de editboxes voor het invullen van de interpolatie parameter
  Stel de zichtbaarheid van de editboxes en de assen in
=====
GEBRUIKTE SUBFUNCTIES:
=====

function showdata(Tculm,Stand,Standint,Havenget,Havengetint,tag,col)
  Plot Tculm tegen Havenget
  Plot Stand tegen Tculm
  Plot Havenget tegen stand (aardappelgrafiek)
  gebruik in alle 3 gevallen:
  'tag'=tag
  'col'=col
  plot indicators voor hele uren in de aardappelgrafiek
  bijbehorende testarray

function showlines(Stand,Standint,Havenget,Havengetint,tag,col)
  plot de volgende indicatie lijnen
  - gemiddelde stand (horizontale lijn, in aardappel grafiek en grafiek stand)
  - snijpunt aardappelgrafiek met gemiddelde stand (vertikale lijn, 2x)
  - gemiddeld haven getal (vertikale lijn)
  plot in de aardappelgrafiek hoogte en uraanduiding bij de lijnen

  gebruik:
  'tag'=tag
  'col'=col

function [curstat,curindex]=getcurstatmenu(data)
  haal naam en index van geselecteerd station op

function data=interp_haven(data,curindex,val)
  interpoleer de curve voor het gemiddelde havengetal
  haal gegevens op uit
  - data(curindex).HavengetHW
  - data(curindex).HavengetLW;
  schrijf gegevens weg naar
  - data(curindex).HavengetHWint
  - data(curindex).HavengetLWint
  De oorspronkelijke, niet geïnterpoleerde gegevens blijven
  dus bewaard, zodat interpolatie desgewenst met andere parameters
  herhaald kan worden
  Voor de feitelijke interpolatie wordt de subfunctie 'interpoleer' aangeroepen

function data=interp_stand(data,curindex,val);
  interpoleer de curve voor de gemiddelde stand
  haal gegevens op uit
  - data(curindex).StandHW
  - data(curindex).StandLW;
  schrijf gegevens weg naar
  - data(curindex).StandHWint
  - data(curindex).StandLWint
  De oorspronkelijke, niet geïnterpoleerde gegevens blijven
  dus bewaard, zodat interpolatie desgewenst met andere parameters
  herhaald kan worden
  Voor de feitelijke interpolatie wordt de subfunctie 'interpoleer' aangeroepen

function y=interpoleer(X,Y,p)
  knoop 3 cycli achter elkaar
  voer interpolatie uit met utility 'cubspline'
  knip middelste gedeelte eruit, dit is de uitvoer
  met het begin en eind van de geïnterpoleerd wordt niets gedaan

function [fheel, uurnrs]=heeluur(Tculm)
  bepaal de indices in Tculm die overeen komen met een uurgemiddelde

```

```
*****
FILE: cubspline.m
```

cubspline - cubic spline approximation with approximate points

```
CALL
  [xout,yout]=cubspline(x0,y0,x,P)
```

```
INPUT
  x0: x coördinaat waargenomen punten
  y0: ycoördinaat waargenomen punten
  x: x coördinaat te schatten punten
  P: gewicht, of diagonaal matrix met gewichten
```

```
OUTPUT
  y: de geëxtrapoleerde punten die horen bij x
```

```
WERKWIJZE
  Uit x0,y0 en P worden de parameters die de interpolatie spline
  karakteriseren bepaald (subfunctie cubspline1)
  Deze parameters volgen uit een aantal matrix bewerkingen die in detail zijn
  gespecificeerd in (de bijlage van) het FO Getijspecials, MobiData, 1999

  Vervolgens vindt extrapolatie uitgevoerd voor punten x
  (subfunctie cubspline2)
  In feite bestaat ge geïnterpoleerde polynoom uit een groot aantal deelpolynomen.
  Voor elk deelpolynoom worden de elementen van x gezocht die tot het relevante
  interval behoren. Vervolgens wordt voor deze waarden de interpolatie bepaald.
```

```
*****
FILE: dispperiode.m
```

dispperiode - Plot periode van overzicht op gestandaardiseerde wijze

```
CALL
  dispperiode(GPDS,T_reeks)
```

```
INPUT
  GPDS: Getijparameterdataset met de volgende kolommen
  kolom1: teken (1 bovensculminatie) (0 onderculminatie) (uit culminatie.mat)
  kolom2: Culminatietijdstip op nulmeridiaan (Greenwich Meant Time) (uit
culminatie.mat)
  kolom3: paralax vd maan (uit culminatie.mat)
  kolom4: declinatie van de maan (uit culminatie.mat)
  kolom5: tijd hoogwater t.o.v. culminatie tijdstip op nulmeridiaan
  kolom6: stand hoogwater
  kolom7: tijd laagwater t.o.v. culminatie tijdstip op nulmeridiaan
  kolom8: stand laagwater
  T_reeks: kolom met tijdstappen (optioneel)
```

```
OUTPUT
  geen
```

```
OUTPUT NAAR SCHERM
  informatie die wordt opgevangen ten behoeve van het overzicht
```

```
WERKWIJZE
  Als invoer bestaat uit GPDS (nargin==1):
  Bepaal de minimum en maximum van kolom2 van GPDS
  Als invoer bestaat uit T_reeks (nargin==2):
  Bepaal de minimum en maximum van T-reeks
  Display begin- en einde van analyseperiode
```

```
OPMERKING
  Kolom 5/7 van GDPS= Tijdstip HW/LW - tijdstip doorgang in plaatselijke
  meridiaan (TDPM)
```

```
*****
FILE: dutchmonth.m
```

dutchmonth - converteer een integer naar een maand (hoofdletters)


```

CALL
  str=dutchmonth(n)

INPUT:
  n: maandnummer

OUTPUT:
  str: textstring met nederlandse maandaanduiding in hoofdletters

WERKWIJZE
  evalueer 'switch n' en ken afhankelijk van n de naam van de maand
  (in hoofdletters) toe aan de string str.

*****
FILE: f_agger.m

  f_agger - bepaal hoog- en laagwaters en aggers

CALL
  NE_reeks=f_agger(E_reeks,T_reeks,N1,N2,P,Pcentre)

INPUT

  E_reeks   : equidistante reeks met standen
  T_reeks   : corresponderende tijdstippen
  N1        : (uren) zoek maximum in intervals t-N1:t+N1
  N2        : (uren) interpoleer met N2 uren
  P         : indien N2>0: gewicht voor afwijking tov waarneming
  Pcentre   : indien N2>0: gewicht voor centrale waarneming

OUTPUT
  NE_reeks: Niet equidistante reeks met de volgende info
    kolom1: tijdstippen
    kolom2:
      code=1==> hoog
      code=2 ==>laag
      code=3 ==> eerste laagwater
      code=4 ==> top agger
      code=5 ==> tweede laagwater
    kolom3: stand

OUTPUT NAAR SCHERM
  informatie die wordt opgevangen ten behoeve van het overzicht

WERKWIJZE
  roep f_hooglaag aan voor bepaling met extremen van leeftijd N1 uur
  stop het resultaat in reeks A
  roep f_hooglaag aan voor bepaling met extremen van leeftijd 90 min
  stop het resultaat in reeks B
  Loop alle laagwaters in A af

  Bepaal voor alle LW's in reeks A (LWA) of er bij deze waarden aggers horen.
  Dit gaat als volgt.

  • stel LWB = de dichtsbijzijnde LW bij LWA in B
  • stel LWB0 = absoluut laagste waterstand in B in interval [LWB-5uur,LWB-2uur]
  • stel HWB0 = absoluut hoogste waterstand in B in interval [LWB0 , LWB]
  • stel LWB1 = absoluut laagste waterstand in B in interval [LWB+2uur,LWB+5uur]
  • stel HWB1 = absoluut hoogste waterstand in B in interval [LWB , LWB1]
  • Ga na of de reeks LWB0 HWB0 LWB aan de jaarboek eis voldoet
  • Ga na of de reeks LWB HWB1 LWB1 aan de jaarboek eis voldoet
  • Indien beide reeksen aan de jaarboekeis voldoen genereer dan een waarschuwing,
    ga door met het minimum van LWB0 en LWB1
  • Als een van de twee reeksen aan de jaarboekeis blijkt te voldoen voer dan
    waarden door in uitvoer met de codes 'eerste laagwater', 'Top agger',
    'tweede laagwater'. Zoniet voer dan het LWB door in de uitvoer.

*****
FILE: f_eqfit.m

  f_eqfit - Fit verloop lijn door equidistante reeks door interpolatie

```

```

CALL
  E_reeks_out=f_eqfit(NE_reeks2,E_reeks,T_reeks,NE_reeks1)

INPUT
  NE_reeks2: Niet equidistante reeks met de volgende info
    kolom1: tijdstippen
    kolom2:
      code=1==> hoog
      code=2 ==>laag
      andere codes mogen niet voorkomen (aggers worden niet geaccepteerd)
  E_reeks: equidistante reeks met standen
  T_reeks: corresponderende tijdstippen

OUTPUT
  E_reeks: equidistante reeks met berekende standen

OUTPUT NAAR SCHERM
  informatie die wordt opgevangen ten behoeve van het overzicht

WERKWIJZE
  doel van de functie:
  Vervormen van een equidistante invoer reeks met de verschil reeks
  NE_reeks2-NE-Reeks1
  Na aflopp van de functie aanroep gaat E-reeks door de extremen in NE-Reeks2'

  Eerst wordt gecontroleerd, of NE_reeks als argument is meegegeven
  Als dit niet het geval is dan wordt NE_reeks aangemaakt door extremen te bepalen
  van de ingevoerde equidistante reeks

  Er wordt voor gezorgd dat NE_reeks1 ne NE_reeks2 op een gelijke of kortere
  periode betrekking hebben dan de ingevoerde equidistante reeks. Hiertoe worden
  alle te vroege of te late elementen uit NE_reeks1 en NE_reeks2 verwijderd.

  NE_reeks1 en NE_reeks2 worden uitgebreid met het eerste datapunt uit de ingevoerde
  equidistante reeks.

  NE_reeks1 en NE_reeks2 worden gesynchroniseerd door alle elementen die niet
  matchen binnen een periode DELTA (ingesteld op 1.5 uur) te verwijderen.
  Wat overblijft zijn de z.g. gemeenschappelijke extremen.
  Het synchroniseren vindt plaats in een aparte procedure synchronize.m

  Er wordt een verschilreeks van samengesteld met twee kanalen:
  NE_reeks1(:,1)-NE_reeks1(:,3) (verschuiving in tijd die nodig is om op
  NE_reeks2 uit te komen)
  NE_reeks2(:,3)-NE_reeks1(:,3) (verschuiving in stand die nodig is om op
  NE_reeks2 uit te komen)

  Deze verschil reeks bevat voor elk tijdstip ui NE_reeks1 twee corresponderende
  verschuivingen.

  Zowel de verschilreeks in tijd als verschilreeks in stand wordt nu lineair
  geïnterpoleerd. Dit gebeurt met behulp van de standaard Matlab routine 'interp1.m'.
  De interpolatie vindt plaats naar de tijdstippen die corresponderen met de
  oorspronkelijk opgegeven equidistante reeks.

  Er wordt een 'hulp' niet equidistante reeks gecreeerd door het verschil in
  tijd en het verschil in stand bij de oorspronkelijke equidistante reeks
  op te tellen.

  Deze hulp reeks wordt weer equidistant gemaakt door lineaire interpolatie.
  Het resultaat hiervan is de uitvoerreeks.

  De gevolgde werkwijze kan worden gevisualiseerd.
  Zet hiervoor de variabele TEST op 1.
  In dat geval wordt een window met de naam 'test' gemaakt met een plot van de
  volgende lijnen:
  oorspronkelijke reeks
  extremen oorspronkelijke reeks
  extremen nieuwe reeks
  tijdelijke NE-reeks
  geïnterpoleerde tijdelijke NE-reeks (de uitvoer reeks)

*****
FILE: f_havenget.m

```

f_havenget - Bereken het gemiddelde havengetal

CALL

```
[TculmHW,StandHW,HavengetHW,TculmLW,StandLW,HavengetLW,begin,einde]=...
  f_havenget(GPDS,deltaT,deltaH,optie,Ptijd,Pstand)
```

INPUT

```
GPDS: Getijparameterdataset met de volgende kolommen
kolom1: teken (1 bovensculminatie) (0 onderculminatie) (uit culminatie.mat)
kolom2: Culminatietijdstip op nulmeridiaan (Greenwich Meant Time)
      (uit culminatie.mat)
kolom3: paralax vd maan (uit culminatie.mat)
kolom4: declinatie van de maan (uit culminatie.mat)
kolom5: tijd hoogwater t.o.v. culminatie tijdstip op nulmeridiaan
kolom6: stand hoogwater
kolom7: tijd laagwater t.o.v. culminatie tijdstip op nulmeridiaan
kolom8: stand laagwater
deltaT: outliercriterium voor tijd binnen klasse (in uren)
deltaH: outliercriterium voor hoogte binnen klasse (in centimeters)
Ptijd: gewicht bij interpoleren Tijd (eenheid: 1/uur^2)
Pstand: gewicht bij interpoleren Stand (eenheid: 1/uur^2)
```

OUTPUT

```
TculmHW   : culminatie tijdstip modulo 12
StandHW   : stand Hoogwater in cm
HavengetHW: periode tussen doorgang door plaatselijke meridiaan en
           eerstvolgende HW
TculmLW   :
StandLW   : stand Laagwater in cm
HavengetLW: periode tussen doorgang door plaatselijke meridiaan en
           eerstvolgende LW
begin     : begin analyseperiode
einde     : einde analyseperiode
```

OUTPUT NAAR SCHERM

informatie die wordt opgevangen ten behoeve van het overzicht

WERKWIJZE

Voer een identieke berekening uit, apart voor Hoog- en Laagwaters
Deze berekening vindt plaats in de functie f_havenget1 en houdt in:

- Verwijder de hiaten die eventueel in de dataset zitten
Deze hiaten zijn te herkennen aan het feit dat NaN is ingevuld voor tijdstip of stand

- verwijder outliers, apart voor boven en onderculminaties
zoek de indices op voor de bovensculminaties (fb) en voor de onderculminaties (fo)
het verwijderen van de outliers is voor de boven- en onderculminaties een identiek proces. Dit proces vindt plaats in de subfunctie F_HAVENGET2.

De werking van dit proces is als volgt:

- alle data worden gesorteerd op basis van tijd van de dag (modulo 24 uur)
- indien optie == 'm' bereken een voortschrijdende mediaan
- indien optie == 'g' bereken een voortschrijdend gemiddelde
opmerking1: Het voortschrijdende kental wordt berekend met een gecentreerde horizon van 1 uur (dis is instelbaar met de constante HORIZON)
opmerking2: voortschrijdende mediaan is minder gevoelig voor outliers dan voortschrijd gemiddelde
- verwijder alle elementen uit de tijdreeks die ten opzichte van het voortschrijdende kental qua hoogte meer dan hoogte deltaH afwijken
- bereken voortschrijdende meridiaan of gemiddelde (afhankelijk van optie) opnieuw
- verwijder alle elementen uit de tijdreeks die ten opzichte van het voortschrijdende kental qua tijd van HW/LW meer dan deltaT afwijken
- voeg boven en onder culminatie samen (in de dataset is nu niet meer te zien of een bepaald HW/LW een boven of onderculminatie betreft)
- voeg culminaties samen modulo een 1/2 dag (in de dataset wordt nu geen onderscheid meer gemaakt tussen bijvoorbeeld 9.00 en 21.00)
- sorteer de resulterende dataset opnieuw op tijd
- bereken opnieuw het gecentreerd voortschrijdend gemiddelde van stand en havengetal. Op basis van deze gegevens kan een de aardappelgrafiek berekend worden
- voor elke uur = 0 tm 12:
voer het voortschrijdende gemiddelde voor tijd en stand uit voor het eerste culminatie tijdstip > uur+1/2 Omdat de straal van de berekening van het voortschrijden gemiddelde steeds een half uur is, worden op deze manier

- steeds uur gemiddelden bepaald
 - Voer tot slot het gemiddelde van de 12 gegevens die hierboven zijn berekend uit

 FILE: f_hooglaag.m

NE_reeks - Bereken hoog- en laagwaters

CALL

NE_reeks=f_hooglaag(E_reeks,T_reeks,N1,N2,P,Pcentre,printopt)

INPUT

E_reeks: equidistante reeks met standen
 T_reeks: corresponderende tijdstippen
 N1: (uren) zoek maximum in intervals t-N1:t+N1
 N2: (uren) interpoleer met N2 uren
 P: indien N2>0: gewicht voor afwijking tov waarneming
 Pcentre: indien N2>0: gewicht voor centrale waarneming
 printopt: (optioneel) print optie
 1==> print regels (default)
 0==> print niets

OUTPUT

NE_reeks: Niet equidistante reeks met de volgende info
 kolom1: tijdstippen
 kolom2: code=1==> hoog code=2 ==>laag
 kolom3: stand

OUTPUT NAAR SCHERM

informatie die wordt opgevangen ten behoeve van het overzicht

WERKWIJZE

De variabele 'printopt' geeft aan of er wel of niet boodschappen naar het scherm moeten worden geschreven. De defaultwaarde is 1 (wel boodschappen op scherm).

N1 en N2 zijn in uren gespecificeerd. Intern rekent de module met periode-aantallen. De periode-aantallen die corresponderen met N1 en N2 uur zijn N1per en N2per.

De maxima en minima van E_reeks worden bepaald met behulp van de utility 'indexmax'. maxperiods en minperiods zijn indices naar maxima en minima met leeftijd >= N1per

Als N2per > 0:

- Interpoleer de maxima met de subfunctie 'interpmax'
- Interpoleer de minima op basis van de negatieve reeks

Als N2per == 0:

- Stel de maxima en minima gelijk aan de gevonden maxima en minima

Zet nu de maxima en minima achter elkaar in een NE-reeks die nog niet op tijd is gesorteerd.

Sorteer deze NE-reeks op tijdstip

Als 'printopt==1' print dan informatie naar het scherm.

=====
 SUBFUNCTIES van f_hooglaag.m
 =====

```
function [maxinstant,maxvalues]=interpmax(maxperiods,x,y,N,P,Pcentre)
    interpoleer maxima
```

INPUT:

maxperiods: indices in x; periodes waar maxima optreden
 x : x coördinaten
 y : y coördinaten
 N : aantal periodes te gebruiken bij interpolatie
 P : interpolatieparameter
 Pcentre : interpolatieparameter centrale waarde

OUTPUT:

maxinstant: de tijdstippen waarop de maxima vallen
 maxvalues : de bijbehorende maxima

De maxima worden bepaald met een 'brute force' methode: De functie wordt geïnterpoleerd naar 1 minuut waarden, waarvan vervolgens het maximum wordt bepaald.

Door het aanpassen van de constante 'STEPSIZE' (momenteel 1 minuut) kan een fijner of grover grid worden gekozen.

In een loop worden alle maxima geïnterpoleerd.

Per maximum:

Bepaal eerste en laatste periode voor interpolatie
 Normaal: $\text{maxperiods}(k) - N, \text{maxperiods}(k) + N$
 Bewaak echter grenzen van array
 strtper is eerste periode, endper is laatste periode.
 (strtper en endper zijn indices in x)
 Verdeel interval strtper-endper in vakjes van lengte STEPSIZE ==> xgrid
 Construeer vectoren X en Y die corresponderen met x(strtper:endper) en y(strtper:endper), maar waaraan links en rechts van het centrum z.g. 'buigpunten' zijn toegevoegd.
 Op deze knikpunten wordt de afwijking niet gewogen, maar wordt wel een discontinuïteit in de tweede afgeleid toegestaan.
 Construeer een matrix P met weegfactoren.

De feitelijke interpolatie wordt uitgevoerd op basis van X, Y en P en levert voor de opgegeven punten xgrid de waardes ygrid op.

De interpolatie vindt plaats in de routine CUBSPLINE

Opm1: Als meerdere maxima worden gevonden wordt het tijdstip gemiddeld.
 Opm2: Als de interpolatie mislukt, verschijnt een melding en wordt de exacte doorgegeven

 FILE: f_laaglaag.m

f_laaglaag - Bereken laag-laagwaterspring

CALL
 f_laaglaag(GPDS)

INPUT

GPDS: Getijparameterdataset met de volgende kolommen
 kolom1: teken (1 bovensculminatie) (0 onderculminatie) (uit culminatie.mat)
 kolom2: Culminatietijdstip op nulmeridiaan (Greenwich Meant Time) (uit culminatie.mat)
 kolom3: paralax vd maan (uit culminatie.mat)
 kolom4: declinatie van de maan (uit culminatie.mat)
 kolom5: tijd hoogwater t.o.v. culminatie tijdstip op nulmeridiaan
 kolom6: stand hoogwater
 kolom7: tijd laagwater t.o.v. culminatie tijdstip op nulmeridiaan
 kolom8: stand laagwater

OUTPUT

geen

OUTPUT NAAR SCHERM

informatie die wordt opgevangen ten behoeve van het overzicht

WERKWIJZE

Bepaal de culminaties die het dichtst bij 0 of 12 liggen.
 Hiertoe wordt een kolom gedefinieerd als het absolute verschil tussen culminatietijdstip en 0 of 12 uur van deze kolom worden minima berekend met een leeftijd van 5 periodes merk op dat de culminaties elke dag +- 50 min opschuiven zodat op deze wijze alle relevante maxima gevonden worden
 Alle hoogwaters die bij de gevonden culminaties horen zijn spring hoogwaters
 De laagwaters die bij deze culminaties horen, en de laagwaters van 1 culminatie terug zijn springlaagwaters
 Voor elk jaar dat geheel of gedeeltelijk in de dataset zit wordt:
 - Van elke maand de minimale stand geplot die bij een spring LW hoort
 - Voor het gehele jaar (voorzover dit in de dataset voorkomt) de gemiddelde LLWS geplot

```

*****
FILE: f_ongelijkheid.m

    f_ongelijkheid - bepaal dagelijkse ongelijkheid

CALL
    f_ongelijkheid(NE_reeks)

INPUT
    NE_reeks: Niet equidistante reeks met de volgende info
              kolom1: tijdstippen
              kolom2: code=1==> hoog code=2 ==>laag
              kolom3: stand

OUTPUT NAAR SCHERM
    informatie die wordt opgevangen ten behoeve van het overzicht

WERKWIJZE:
    Dagelijkse ongelijkheid voor waarneming i:
      = abs(HW(i)-HW(i+1)) indien 6:00<=HW(i)<18:00 en 18:00<=HW(i+1)<6:00
      of      18:00<=HW(i+1)<6:00 en 6:00<=HW(i)<18:00

    Splits de NE reeks in een gedeelte met alleen Hoogwaters en een gedeelte met
    alleen laagwaters
    roep voor beide gedeeltes de functie 'ongelijkheid_subfunc' aan

    functie ongelijkheid_subfunc:
    INPUT
        NE_reeks: Niet equidistante reeks met de volgende info
                kolom1: tijdstippen
                kolom2: code=1==> hoog code=2 ==>laag
                kolom3: stand

    OUTPUT
        ongelijkheid      : gemiddelde dagelijkse ongelijkheid
        maxongelijkheid  : maximum dagelijkse ongelijkheid
        succes            : =1 als berekening succesvol

    Werkwijze:
    - Bepaal tijd van de dag in uren
    - Bepaal alle Extremen met 6:00<=extreem(i)<18:00 en 18:00<=extreem(i+1)<6:00
      Controleer ook dat de extremen niet verder dan 24uur verwijderd liggen ivm hiaten
    - Bepaal alle Extremen met 18:00<=extreem(i)<6:00 en 6:00<=extreem(i+1)<18:00
      Controleer ook dat de extremen niet verder dan 24uur verwijderd liggen ivm hiaten
    - Definieer delta als de verschillen tussen twee elkaar opvolgende extremen

*****
FILE: f_tussenschr.m

    f_tussenschr - Bepaal tussenschreiding

CALL
    f_tussenschr(E_reeks,T_reeks,ondergrens,bovengrens,printoptie)

INPUT
    E_reeks: equidistante reeks met standen
    T_reeks: corresponderende tijdstippen
    ondergrens: onder grens voor de standen van de te zoeken periodes
    bovengrens: boven grens voor de standen van de te zoeken periodes
    printoptie:
        0 - print alleen geaggregeerde kentallen
        1 - print begin en eind van elke periode
        2 - print begin en aantal tijdstappen van elke periode

OUTPUT
    geen

OUTPUT NAAR SCHERM
    informatie die wordt opgevangen ten behoeve van het overzicht

WERKWIJZE
    Bepaal op basis van het criterium, voor welke periode tussenschrijding
    plaatsvindt.
    Voer het percentage tussenschrijding uit naar het scherm.
    Groepeer de tussenschrijdingsperiodes in clusters van aaneensluitende
    periodes dit gebeurt in de functie GROUP.
    Print voor elke groep een regel op het scherm (behalve als printoptie==0)
    De opmaak van de regel hangt af van de invoer variabele 'printoptie'.

```

```
*****
FILE: f_vliegplan.m

f_vliegplan - Bepaal onderschreiding

CALL
function
f_vliegplan(E_reeks,T_reeks,bovengrens,kustvaknaam,stationsnaam,vliegrichting,deltat)

INPUT
E_reeks      : equidistante reeks met standen
T_reeks      : corresponderende tijdstippen
bovengrens   : boven grens voor de standen van de te zoeken periodes
kustvaknaam  : aanduiding kustvak
stationsnaam : aanduiding station
vliegrichting: aanduiding vliegrichting
deltat       : vervroeging van de vliegperiode ten opzichte van
              berekende periode
```

```
*****
FILE: f_zuiver.m

f_zuiver - zuiver aggers

CALL
NE_reeks=f_zuiver(NE_reeks,optie)

INPUT
NE_reeks: Niet equidistante reeks met de volgende info
kolom1: tijdstippen
kolom2:
code=1==> hoog
code=2 ==>laag
code=3 ==> eerste laagwater
code=4 ==> top agger
code=5 ==> tweede laagwater
kolom3: stand
optie: uit te voeren actie
optie=1 ==>LW agger vervangen door laagste LLW met tijd en code 2
optie=2 ==>controleer aggers op jaarboekjes, als ze niet
voldoen vervangen LLW
optie=3 ==>Handhaaf LW1 met code 2, verwijder LW2
optie=4 ==>Handhaaf LW2 met code 2, verwijder LW1
optie=5 ==>Voer een aantal logische controles uit
(code 3 wordt niet door code 4 en 5 gevolgd door laagste LW)

OUTPUT
NE_reeks: Niet equidistante reeks met de volgende info
kolom1: tijdstippen
kolom2:
code=1==> hoog
code=2 ==>laag
code=3 ==> eerste laagwater
code=4 ==> top agger
code=5 ==> tweede laagwater

kolom3: stand

OUTPUT NAAR SCHERM
informatie die wordt opgevangen ten behoeve van het overzicht

WERKWIJZE
Als de opgegeven code gelijk is aan 1,2 ,3 of 4 (zie boven), dan is de werkwijze
als volgt:
Eerst worden alle top aggers opgespoord in de te bewerken dataset.
Vervolgens wordt gekeken of het inderdaad om een complete agger gaat.
Dat wil zeggen dat het voorgaande record in de NE reeks een LW1 (code 3)
moet zijn en het volgende record in de NE reeks een LW2 (code 5) moet zijn.
Indien een van beide niet het geval is wordt de volgende melding gegeven:

>> <periodeaanduiding>: incomplete agger
```

Er wordt voor deze agger dan geen actie uitgevoerd.

- Als de agger compleet is dan hangt de verdere actie af van de opgegeven code:
1. LW agger wordt vervangen door laagste LW, met de bijbehorende tijd en code 2.
 2. Vervang aggers die niet aan de jaarboekeis voldoen door het laagste LW.
 3. Handhaaf LW1 met code 2 en verwijder LW2.
 4. Handhaaf LW2 met code 2 en verwijder LW1
 5. Voer een aantal logische controles uit.

FILE: filterandsmooth.m

filterandsmooth - Verwijder outliers en bereken gemiddelde (voor verwijdering outliers).

Ook geschikt voor cyclische reeksen.

Outlierverwijdering gebeurt op basis van vergelijking met gecentreerd voortschrijdende mediaan of gemiddelde.

CALL

[f,ysmooth]=filterandsmooth(x,y,HORIZON,OUTLIERCRIT,optie,modulo)

INPUT

x : x coördinaten reeks
y : y coördinaten reeks
HORIZON : totale lengte middelings horizon (verwijst naar x)
(de straal van de middelings horizon is dus HORIZON/2)
OUTLIERCRIT: criterium voor verwijderen outliers (verwijst naar y)
optie : berekeningswijze
'g' : neem gemiddelde
'm' : neem median
modulo : (optioneel) tijdsduur periode bij cyclische reeksen (verwijst naar x)
(eind zit aan begin vast, bijvoorbeeld etmaal)
Opmerking: Als de variabele modulo is gedefinieerd, wordt aangenomen dat de variabele x het interval [0,modulo] beslaat.
Opmerking: Modulo moet groter dan max(x) zijn, anders is uitvoer niet zinnig.

OUTPUT

f : indices (in x en y) van elementen die mogen blijven
ysmooth : bijbehorende gladgestreken y waarden

WERKWIJZE

Als de invoer variabele modulo een waarde heeft, dan:
- Worden aan 'first' de indices toegekend van de elementen $X \leq \text{HORIZON}/2$
- Worden aan 'last' de indices toegekend van de elementen $X \geq \text{modulo} - \text{HORIZON}/2$

De variabelen xx (en yy) worden gedefinieerd als:
De laatste elementen van x (y), x (y), de eerste elementen van x (y)

Als optie=='g' worden van xx en yy het moving average berekend in de submodule MOVINGAVERAGE

Als optie=='m' worden van xx en yy de moving median berekend in de submodule MOVINGMEDIAN

Uit het resultaat wordt weer het gedeelte geknipt dat correspondeert met x en y

Vervolgens wordt dit met y vergeleken, en getoetst aan het outliercriterium

De uitvoer bestaat uit de indices (f) van de elementen van y die aan het outliercriterium voldoen en het berekende voortschrijdend gemiddelde (ysmooth).

FILE: getij.m

getij - activeren getijspecial menu's

CALL


```

getij

INPUT
geen

OUTPUT
geen

OUTPUT NAAR SCHERM
er worden een aantal menu's geactiveerd

WERKWIJZE
De hidden handles van de Melissa interface worden zichtbaar gemaakt
De handle van het mainmenu wordt opgespoord
Er worden een aantal menu's toegevoegd
De handles van de Melissa worden weer in hun oude staat hersteld

```

```

*****
FILE: getstatuscodes.m

```

```

getstatuscodes - compileerbare versie van statuscodes.m

CALL
s=getstatuscodes

INPUT
geen

OUTPUT
s: structure met DONAR en Melissa statuscodes

WERKWIJZE
vul de structure met constanres

```

```

*****
FILE: group.m

```

```

group - deel x in groepen in op basis van onderlinge afstand van elementen van x
in elke groep mag de afstand tussen twee opeenvolgende elementen
niet groter dan maxstep zijn

CALL
xgr=group(x,maxstep)

INPUT
x      : te clusteren reeks
maxstep: de maximale stapgrootte (verwijst naar x)

OUTPUT
xgr: per element van x een getal dat aangeeft in welke groep dit
element zit

WERKWIJZE
voorbeeld:
group([1 2 3 5 6],1)=[1 1 1 2 2];

De achterliggende manier van berekenen van dit voorbeeld bestaat
uit de volgende stappen:
x=      1 2 3 5 6 7 8
diff(x)= 1 1 2 1 1 1

Uitbreiden met 0 aan het begin:
y=      0 0 0 1 0 0 0
Het resultaat na het nemen van een cumulatieve som.
cumsum(y)+1= 1 1 1 2 2 2 2

```

```

*****
FILE: indexmax.m

```

```

indexmax - Bepaal maxima in reeks x met leeftijd N

```

```

CALL
  [maxindices,minindices]=indexmax(x,N)

INPUT
  x: vector met potentiële maxima (minima)
  N: minimale leeftijd van het gecentreerde maximum (verwijst naar index in x)

OUTPUT
  maxindices: de indices van de maxima
  minindices: de indices van de minima

WERKWIJZE
  Construeer matrix tempMat met verschoven kolommen:
    0   0   x2  x3
    0  x1  x3  x4
  x1  x2   x4
  x2  x3
  x3
  Roep de submodule GETMAX aan om de maxima van leeftijd N op te sporen
  Roep de submodule GETMAX aan om de minima van leeftijd N op te sporen

=====
SUBMODULES van indexmax.m
=====

function maxindices=getmax(tempmat,x,N)
  spoor maxima op
  garandeer een leeftijd van N periodes
INPUT
  tempmat: geconstrueerde matrix met verschoven kolommen
  x       : reeks waarvan maxima bepaald moeten worden
  N       : leeftijd van gezochte maxima (verwijst naar indices in x)
OUTPUT
  maxindices: indices van gevonden maxima

  Bepaal de rij maxima van tempmat.
  Stel de eerste N elementen op inf (zodat x deze niet zal overtreffen).
  Stel de laatste N elementen op inf (zodat x deze niet zal overtreffen).
  Bepaal de elementen van x die maximaal zijn met leeftijd N

N.B:
  De bovenstaande procedure vindt waarden in de vector x die
  groter OF GELIJK zijn aan waarden in interval x(tmax-N), x(tmax+N)
  indien het gelijk teken geldt moet een van beide verwijderd worden
  In dit geval: vervang door (n+1)/2 e exemplaar (naar beneden afronden)
  Dit gebeurt in de submodule ZEEF

function indices_zeef=zeef(indices,N)
  vervang clusters van periodes met maximale afstand N door middelste waarde
INPUT
  indices: indices van kandidaat maxima
  N      : vereiste leeftijd van maxima

OUTPUT
  indices_zeef: subset van indices die vereiste leeftijd heeft

VOORBEELD
  als N=2, dan is reeks [4 7] correct,
  maar reeks [4 6] wordt vervangen door [4]

WERKWIJZE
  Deel de indices in groepen waarvan de elementen een maximale onderlinge
  afstand
  N+.1 hebben (afstand N is dus toegestaan).
  Dit gebeurt met behulp van de functie GROUP

  Bepaal per groep het centrale element (na beneden afgerond):
    1=>1
    2=>1
    3=>2 etc
  Sla dit element op in de uitvoervariabele indices_zeef

*****
FILE: mls_agger.m

```

mls_agger - User interface voor de bepaling van hoog en laag water en aggers

CALL

mls_agger(fname)

INPUT

fname: kan een waarde hebben als functie in batch mode wordt aangeroepen
 fname wordt niet gebruikt
 als fname een waarde heeft (nargin==1) wordt het overzicht alleen
 weggeschreven als file, en niet geactiveerd.

OUTPUT

De property 'userdata' wordt aangepast: er wordt een NE-reeks naar toe geschreven

WERKWIJZE

De berekening wordt voor ieder station apart uitgevoerd.
 Tijdens de berekening wordt de melissa interface gedeactiveerd
 voor alle stations met code 'berekenen==aan wordt functie f_agger aangeroepen.
 Het resultaat van deze aanroep wordt tijdelijk opgeslagen in het cell array
 'cell_data'. De current index van van dit cell array wordt bijgehouden in de
 integer
 cell_index, die ook gebruikt wordt voor het plotten van een wachtbalk.
 De stationsnummers waarvoor berekeningen zijn uitgevoerd worden opgeslagen in het
 array 'cell_statnrs'.
 Als de berekening succesvol is verlopen worden alle gegevens in 1 keer naar het
 werkgebied weggeschreven.

FILE: mls_eqfit.m

mls_eqfit - User interface voor het fitten van een equidistante reeks op een
 niet-equidistante reeks

CALL

mls_eqfit(fname)

INPUT

fname: kan een waarde hebben als functie in batch mode wordt aangeroepen
 maar wordt niet gebruikt
 als fname een waarde heeft (nargin==1) wordt het overzicht alleen
 weggeschreven als file, en niet geactiveerd.

OUTPUT

De property 'userdata' wordt aangepast: er wordt een NE-reeks naar toe geschreven

WERKWIJZE

De berekening wordt voor ieder station apart uitgevoerd.
 Tijdens de berekening wordt de melissa interface gedeactiveerd
 voor alle stations met code 'berekenen==aan wordt functie f_eqfit aangeroepen.
 Het resultaat van deze aanroep wordt tijdelijk opgeslagen in de matrix E_reeks
 De current index van van deze matrix wordt bijgehouden in de integer
 cell_index, die ook gebruikt wordt voor het plotten van een wachtbalk.
 De stationsnummers waarvoor berekeningen zijn uitgevoerd worden opgeslagen in het
 array 'cell_statnrs'.
 Als de berekening succesvol is verlopen worden alle gegevens in 1 keer naar het
 werkgebied weggeschreven.

FILE: mls_expgdps.m

mls_expgdps(fname) - User interface voor exporteren GDPS

CALL

mls_expgdps(fname)

INPUT

fname: kan een waarde hebben als functie in batch mode wordt aangeroepen

OUTPUT

Er wordt een mat file met naam fname weggeschreven

WERKWIJZE

Bepaal de stations met code berkenen

Bewaar voor deze stations de getijparameterset indien deze niet leeg is

FILE: mls_export.m

mls_export - Exporteer (een gedeelte van) de Melissa Werkgebied opslagstructuur

CALL

out=mls_export(fldnm)

INPUT

fldnm: leeg of afwezig: exporteer de hele database
specifieke naam: exporteer het veld met deze naam

OUTPUT

out: data uit de Melissa Werkgebied opslagstructuur

WERKWIJZE

Haal de Melissa opslagstructuur op.
Als er geen invoerargument is meegegeven (nargin==0):
Exporteer de gehele structure.
Als er wel een invoer argument is meegegeven:
Controleer of het invoerargument een string is, zo ja:
Controleer of het invoerargument een geldige naam is, zo ja:
Exporteer structure.veldnaam
zo nee:
Display een list van geldige velnamen

VOORBEELD

```
u=melissa('export');plot(u.W,u.V);
of identiek:
W=melissa('export','W');
V=melissa('export','V');
plot(W,V);
```

FILE: mls_havenget.m

mls_havenget - User interface voor de bepaling van havengetallen

CALL

mls_havenget(fname)

INPUT

fname: kan een waarde hebben als functie in batch mode wordt aangeroepen
fname wordt niet gebruikt
als fname een waarde heeft (nargin==1) wordt het overzicht alleen
weggeschreven als file, en niet geactiveerd.

OUTPUT

De property 'userdata' wordt aangepast: er wordt een NE-reeks naar toe geschreven

WERKWIJZE

Uit een ascii input file wordt de stuurparameters gelezen
Deze worden tijdelijk opgeslagen in verschillende arrays
Tijdens de berekening wordt de melissa interface gedeactiveerd
Voor alle stations met code 'berekenen==aan wordt functie f_havenget aangeroepen
de uitvoer van deze functie wordt tijdelijk opgeslagen in structure array 'data'
Als de berekening succesvol is verlopen, wordt het structure array doorgegeven aan
een zelfstandige GUI: AARDAPPELS. In deze GUI kan dan de aardappelgrafiek
worden bekeken. Er worden geen gegevens naar het werkgebied weggeschreven.

FILE: mls_hooglaag.m

mls_hooglaag - User interface voor de bepaling van hoog en laag water

CALL

mls_hooglaag(fname)

INPUT

fname: kan een waarde hebben als functie in batch mode wordt aangeroepen
 fname wordt niet gebruikt
 als fname een waarde heeft (nargin==1) wordt het overzicht alleen
 weggeschreven als file, en niet geactiveerd.

OUTPUT

De property 'userdata' wordt aangepast: er wordt een NE-reeks naar toe geschreven

WERKWIJZE

Uit een ascii input file wordt de stuurparameters gelezen.
 Deze worden tijdelijk opgeslagen in verschillende arrays.
 Tijdens de berekening wordt de melissa interface gedeactiveerd
 De berekening wordt voor ieder station apart uitgevoerd.
 voor alle stations met code 'berekenen==aan wordt functie f_hooglaag aangeroepen.
 Het resultaat van deze aanroep wordt tijdelijk opgeslagen in het cell array
 'cell_data'. De current index van van dit cell array wordt bijgehouden
 in de integer cell_index, die ook gebruikt wordt voor het plotten van
 een wachtbalk.
 De stationsnummers waarvoor berekeningen zijn uitgevoerd worden opgeslagen in het
 array 'cell_statnrs'.
 Als de berekening succesvol is verlopen worden alle gegevens in 1 keer naar het
 werkgebied weggeschreven.

FILE: mls_laaglaag.m

mls_laaglaag - User interface voor de bepaling van laag laagwaterspring

CALL

mls_laaglaag(fname)

INPUT

fname: kan een waarde hebben als functie in batch mode wordt aangeroepen
 fname wordt niet gebruikt
 als fname een waarde heeft (nargin==1) wordt het overzicht alleen
 weggeschreven als file, en niet geactiveerd.

OUTPUT

geen

WERKWIJZE

tijdens de berekening wordt de melissa interface gedeactiveerd
 voor alle stations met code 'berekenen==aan wordt functie f_laaglaag aangeroepen

FILE: mls_ongelijkheid.m

mls_ongelijkheid - User interface voor de bepaling van de dagelijkse ongelijkheid

CALL

mls_ongelijkheid(fname)

INPUT

fname: kan een waarde hebben als functie in batch mode wordt aangeroepen
 fname wordt niet gebruikt
 als fname een waarde heeft (nargin==1) wordt het overzicht alleen
 weggeschreven als file, en niet geactiveerd.

OUTPUT

Er wordt alleen gegevens naar een overzicht weggeschreven

WERKWIJZE

Tijdens de berekening wordt de melissa interface gedeactiveerd
 Voor alle stations met code 'berekenen==aan wordt functie
 f_ongelijkheid aangeroepen

FILE: mls_tussenschr.m

mls_tussenschr - User interface voor de bepaling van de tussenschrijding

CALL
 mls_tussenschr(fname)

INPUT
 fname: kan een waarde hebben als functie in batch mode wordt aangeroepen
 fname wordt niet gebruikt
 als fname een waarde heeft (nargin==1) wordt het overzicht alleen
 weggeschreven als file, en niet geactiveerd.

OUTPUT
 Er wordt alleen uitvoer naar een overzicht geschreven.

WERKWIJZE
 Eerst worden de stuurparameters uit een invoerfile gelezen.
 Tijdens de berekening wordt de melissa interface gedeactiveerd.
 Voor alle stations met code 'berekenen==aan' wordt functie f_tussenschr aangeroepen.

 FILE: mls_vliegplan.m

mls_vliegplan - User interface voor de bepaling van het vliegplan

CALL
 mls_vliegplan(fname)

INPUT
 fname: kan een waarde hebben als functie in batch mode wordt aangeroepen
 fname wordt niet gebruikt
 als fname een waarde heeft (nargin==1) wordt het overzicht alleen
 weggeschreven als file, en niet geactiveerd.

OUTPUT
 Er wordt alleen uitvoer naar een overzicht geschreven.

WERKWIJZE
 Eerst worden de stuurparameters uit een invoerfile gelezen.
 Tijdens de berekening wordt de melissa interface gedeactiveerd.
 Voor alle stations met code 'berekenen==aan' wordt functie f_vliegplan aangeroepen.

 FILE: mls_zuiver.m

mls_zuiver -User interface voor het zuiveren van aggers

CALL
 mls_zuiver(fname)

INPUT
 fname: kan een waarde hebben als functie in batch mode wordt aangeroepen
 fname wordt niet gebruikt
 als fname een waarde heeft (nargin==1) wordt het overzicht alleen
 weggeschreven als file, en niet geactiveerd.

OUTPUT
 De property 'userdata' wordt aangepast: er wordt een NE-reeks naar toe geschreven

WERKWIJZE
 De berekening wordt voor ieder station apart uitgevoerd.
 Tijdens de berekening wordt de melissa interface gedeactiveerd
 Met behulp van het commando TEXTREAD wordt een parameter uit een ascii file gelezen
 voor alle stations met code 'berekenen==aan' wordt functie f_zuiver aangeroepen.
 Het resultaat van deze aanroep wordt tijdelijk opgeslagen in het cell array
 'cell_data'. De current index van van dit cell array wordt bijgehouden in de
 integer cell_index, die ook gebruikt wordt voor het plotten van een wachtbalk.
 De stationsnummers waarvoor berekeningen zijn uitgevoerd worden opgeslagen in het
 array 'cell_statns'. Als de berekening succesvol is verlopen worden alle
 gegevens in 1 keer naar het werkgebied weggeschreven.

 FILE: pix2norm.m

```

pix2norm - verschuif x en y shiftx en shifty pixels

CALL
  [X,Y]=pix2norm(x,y,shiftx,shifty)

INPUT
  x      : x-coördinaat in genormaliseerde eenheid (max ook vector zijn)
  y      : y-coördinaat in genormaliseerde eenheid (max ook vector zijn)
  shiftx : verschuiving in pixels
  shifty : verschuiving in pixels

INPUT (OVERIGE):
  de 'xlim' en 'ylim' property van de current axis

OUTPUT
  X      : verschoven x-coördinaat in genormaliseerde eenheid
  Y      : verschoven y-coördinaat in genormaliseerde eenheid

WERKWIJZE:
  Sla 'units' property van current axis op in variabele units.
  Verander de 'units' property van current axis naar 'pixels'
  Haal de positievector op (in pixels)
  Haal xlim en ylim op (in normalized coordinaten)
  Bereken X en Y

*****
FILE: synchronize.m

  synchronize - utility voor synchroniseren van twee reeksen
                bepaal corresponderende elementen, gooi overige weg

CALL
  [f1,f2]=synchronize(t1,t2,D)

INPUT
  t1,t2: gesorteerde, te synchroniseren reeksen
  D     : maximaal toegestane afstand tussen corresponderende punten

OUTPUT
f1,f2 : indices in t1 en t2 zdd:
        length(f1)=length(f2)
        max(abs(t1(f1)-t2(f2)))<D | f1=[]

WERKWIJZE
  t1 en t2 worden gesynchroniseerd door alle elementen die niet matchen binnen een
  periode D te verwijderen. Wat overblijft zijn twee indexreeksen van gelijke lengte
  waarvoor geldt:
        max(abs(t1(f1)-t2(f2)))<D | f1=[]
  Bepaal steeds het eerste element waarvoor geen match optreedt
  Afhankelijk van welk element het kleinste is, laat uit een van beide reeksen een
  element weg.
  Als alle (overgebleven) elementen matchen kan gestopt worden.

*****
FILE: tijd2zomertijd.m

  t=tijd2zomertijd(t) - converteer tijdmatrix naar matrix die
                       rekening houdt met zomertijd

CALL
  [t,zomertijd]=tijd2zomertijd(t)

INPUT
  t: matrix met datenum tijdaanduiding
    als t=='compute': display een string die naar deze functie
    gekopieerd kan worden. Dit is nodig indien zomertijd conventies
    veranderen.

OUTPUT
  t: matrix met datenum tijdaanduiding, rkenening houdende met zomertijd.
  zomertijd: matrix met afmetingen van t zomertijd(i,j)=1 als t(i,j) valt in zomertijd
            periode

OUTPUT NAAR SCHERM

```

door `t=='compute'` als argument mee te geven
kan een uitvoer string op het scherm geplot worden die weer in programma
gekopieerd kan worden (eventueel na aanpassing)

WERKWIJZE

Voor elk element van de matrix `t`:
Bepaal het jaar van de laatste start van de zomertijd.
Bepaal wanneer de zomertijd in dat jaar afloopt.
Vergelijk dit met het huidige element van `t`.
Indien zomertijd van toepassing, tel dan 1 uur bij element op.

Opmerking 1: de matrix `'Tbl'` bevat voor elk jaar het begin en einde
van de zomertijd IN STANDAARD TIJDEENHEID (dus zonder correctie
voor zomertijd)

Opmerking 2: Deze code is niet gevectoriseerd. Een versnelling
kan desgewenst bewerkstelligd worden door optimalisatie van de code.
Voor de huidige toepassing is dit echter niet interessant.