User's Manual
TIDEMAT (TIDEGUI)

TIDEMAT tidal analysis

Rijkswaterstaat
Ministerie van Infrastructuur en Waterstaat

User's Manual TIDEMAT

TIDEMAT tidal analysis

Version	:	1.05, September, 2012
Maintenance	:	see www.helpdeskwater.nl/waqua
Copyright	:	Rijkswaterstaat

Versie-geschiedenis

versie	datum	JIRA	Wijzigingen ten opzichte van de vorige versie
1.05	11-09-2012	3700	Diverse tekstuele verbeteringen

Client Rijkswaterstaat RIKZ

Title **TIDEMAT tidal analysis**
Users manual

Abstract

This report describes the design of a toolkit for tidal analysis of series of observations of waterlevels. The toolkit consists of a set of functional routines which can be used individually as so-called Matlab scripts. Also a simple Graphical User Interface is available to support the use of the functional routines and for visualising the series and the components. The software has been build in MATLAB 6.1 software and is applicable on NT as well as Windows 95/98/2000/XP operating systems, provided that the MATLAB 6.1 software is available. It is expected that the software works as well in a Linux environment with MATLAB 6.1.

References

Rev.	Originator		Date	Remarks	Checked by		Approved by	
1			09-12-01	1.0			P. van den Bosch	
1.01	C. van Velzen (VORtech)		14-01-04					

Document Control		Contents		Status
Report number:	A872R1r1	text pages:	13	☐ preliminary
Keywords:	Tide, Graphical User Interfaces , Tidal analysis	tables:	0	☒ draft
Project number:	A872	figures:	0	☐ final
File location: Q:\trunk\src\matlab\tidegui\doc\usedoc\tidegui_users_manual.doc1		appendices:	4	

Alkyon
Hydraulic Consultancy & Research

Executive's summary

This report describes the design of a toolkit for tidal analysis of series of observations of waterlevels. The toolkit consists of a set of functional routines which can be used individually as so-called MATLAB scripts. Also a simple Graphical User Interface is available to support the use of the functional routines and for visualising the series and the components. The software has been build in MATLAB 6.1 software and is applicable under NT and Windows 95/98/2000/XP operating systems on PC's as well as on workstations, as long as the MATLAB software is available. It is expected that the software works as well in a Linux environment with MATLAB 6.1.

No knowledge of MATLAB is necessary to use the GUI. The GUI has a windows look and feel, which is completely in agreement with the look and feel of most of the windows based software. The intuition of the user guides him through the user interface. Exporting figures or plots in postscript, or copying plots or figures towards the clipboard for reporting purposes in Word documents is as easy as any other copy action under windows.

In the design philosophy care has been taken for a quick and efficient use of data. Automatic scaling has been used almost everywhere, for quick handling.

The design philosophy behind the GUI is framed in a data-structure, which is following as much as possible the MATLAB structure. At the same time as less as possible use was made of the global variable space in the MATLAB data structure. This opens the possibilities to use the MATLAB command language independent of TideMat or within TideMat, as desired by the user.

Contents

1 Introduction

1.1 Background

Within the scope of the project NAUTILUS, "Rijkswaterstaat RIKZ" is trying to improve the operational prediction of waterlevels along the coast. An important aspect in generating these predictions is the astronomical tide. The astronomical tide as analysed and predicted from a series of observations of several years for the tide tables of the Netherlands, appears to be incomplete. These incompleteness are especially felt in the actual short term predictions. In order to predict the waterlevel as best as possible it is desirable to improve the prediction accuracy. One of the possibilities to do this is the improvement of the astronomical tide.
A simple toolkit for harmonical and fourier analysis of series of observations is required, because it is expected that good results for short term predictions can be achieved with an adaptive harmonical analysis . This was the reason for Rijkswaterstaat RIKZ to write a request for proposal for the design and construction of a set of modules, including a graphical user-interface, to perform simple analysis with respect to the astronomical tide.

On 17 August 2001, Rijkswaterstaat RIKZ in The Hague, requested Alkyon Hydraulic Consultancy and Research at Emmeloord, by a letter (in Dutch) with reference RIKZ/2001/06162 to submit such a proposal.
Such a final proposal (in Dutch) was submitted to Rijkswaterstaat RIKZ on August 31, 2001 with reference A872le01/GvB.
The assignment was given on 03 October, 2001 by letter with assignment nr. 71011493.
The assignment resulted in version 1.00 of the TideMat software and the documentation which is the present document.
The software modules consists of the following elements:

- Import- and export of data (sds2obs and sds2mat format).
- Selection of tidal components
- Transformation from Fourier components to series and backward.
- Conversion to astronomical arguments and backward
- Dividing of components
- Corrections due to nodal factors
- Filter operations for series and components
- Visualising of series, components and spectra
- All elements callable by script and by user-interface

The software modules can be used stand alone and within the graphical user-interface.

2 Functional modules

The toolkit consists of a number of basic modules. Each of these modules can be used as Matlab script. All the data used by the modules is stored in the so-called tide struct. Each data object of series or components in the tide struct is referenced by an unique label. This label must be defined by the user, when the data object is created. This is a Matlab struct, which is initialised by the 'InitializeTide' module, which always must be the first called module in a script session. The modules which can be used in a script session are:

- InitializeTide - initialize tidal analysis.
- ImportSeries - import series data.
- ImportComponents - import component data.
- Series2Components - transform series to components.
- Components2Series - transform components to series.
- Series2Fourier - FFT transformation of series.
- Fourier2Series - inverse FFT transformation.
- FilterElement - filter Fourier components or series.
- ExportSeries - export series data.
- ExportComponents - export component data.
- RemoveElement - remove data (series or components) from memory.
- AddElements - add operation for two data elements (series or components).
- SubtractElements - subtract operation for two data elements (series or components).
- SetTimeLabels - set labels and ticks along a time axis.
- ShowTide - plot series, components and spectra.

Each of the functions above has the tide struct as input and as output argument. For a description of the tide struct see Appendix D.

2.1 Description of input and output arguments

A short description of each of these functions is given below.

InitializeTide

Purpose:	Initialize the datastructure tide and set default values.		
Call:	tide = InitializeTide;		
Input:	none.		
Output:	tide	– overall struct tide computation with Tdata struct.	

ImportSeries

Purpose:	Import series data from file. Both Obs2Sds format and Sds2Mat file formats can be choosen.	
Call:	tide = ImportSeries(tide, datafile, station, format, label);	
Input:	tide	– tide struct, with Tdata.
	datafile	– file specification of datafile.
	station	– name of station. Default: first station in file.
	format	– format of datafile ('SDS2MAT' or 'OBS2SDS').
	label	– name of new entry in Series() list. Default: generated from station name.
	refdate	– reference date (only for SIMINP format)
	all_stations	– read all stations from input (1/0) (only for SIMINP format)
Output:	tide	– tide struct with new entry in Series() list.

ImportComponents

Purpose: Import astronomical components from file. Both Obs2Sds format and Sds2Mat file formats can be choosen.

Call: tide = ImportComponents(tide, datafile, station, format, label);

Input:
- tide – tide struct, with Tdata.
- datafile – file specification of datafile.
- station – name of station. Default: first station in file.
- format – format of datafile ('SDS2MAT', 'OBS2SDS' or 'SIMINP').
- label – name of new entry in Comp() list. Default: generated from station name.
- refdate – reference date (only for SIMINP format)
- all_stations – read all stations from input (1/0) (only for SIMINP format)

Output: tide – tide struct with new entry in Comp() list.

Series2Components

Purpose: Harmonic analysis of series into components.

Call: tide = Series2Components(tide, serieslabel, set, complabel, time, trncomp, compSplit, Scomp);
or:
tide = Series2Components(tide, serieslabel, set, complabel, time, trncomp);
tide = Series2Components(tide, serieslabel, set, complabel, time);
tide = Series2Components(tide, serieslabel, set, complabel);

Input:
- tide – tide struct with Tdata and Series().
- seriesabel – name of entry in Series() list to analyse.
- set – name of predefined set of constituents, or array with constituent names.
- complabel – name of entry in Comp() list.
- time – start time, end time of timeseries part to analyse. Default: whole series.
- trncomp – Transform components to 1-1-1900? yes/no – 1/0; default no.
- compSplit – Component separation included yes/no – 1/0; default no.
- Scomp{} – Component names to separate.

Output: tide – tide struct with new entry in Comp() list.

Components2Series

Purpose: Prediction of series from a set of astronomical components.

Call: tide = Components2Series(tide, complabel, refdates, serieslabel, timedomain, incr, compSplit, Scompl());
or:
tide = Components2Series(tide, complabel, refdates, serieslabel, timedomain, incr);

Input:
- tide – tide struct with Tdata, Series() and Comp.
- complabel – name of entry in Comp() to analyse.
- refdate – reference dates for prediction.
- serieslabel – name of new entry in Series() list.
- Timedomain - start and end date/time for prediction.
- incr - increment in minutes.
- compSplit – Component splitting included yes/no – 1/0; default no.

Scomp{} – Component names to split.
Output: tide – tide struct with new entry in Series() list.

Series2Fourier
Purpose: Transform the series to fourier components with FFT.
Call: tide = Series2Fourier(tide, serieslabel, complabel, time);
or:
tide = Series2Fourier(tide, serieslabel, complabel);
Input: tide – tide struct with Tdata and Series().
seriesabel – name of entry in Series() list to analyse.
complabel – name of entry in Comp() list.
time – start time, end time of timeseries part to analyse. Default: whole series.
Output: tide – tide struct with new entry in Comp() list.

Fourier2Series
Purpose: Transform fourier components to series with inverse FFT.
Call: tide = Fourier2Series(tide, complabel, serieslabel);
Input: tide – tide struct with Tdata, Series() and Comp.
complabel – name of entry in Comp() to analyse.
serieslabel – name of new entry in Series() list.
Output: tide – tide struct with new entry in Series() list.

FilterElement
Purpose: Filter Fourier components or series.
Call: tide = FilterElement(tide, label, filtertype, domain, newlabel, time);
or:
tide = FilterElement(tide, label, filtertype, domain, newlabel);
Input: tide – tide struct with Tdata, Series() and Comp.
label – name of entry in Comp() or Series() to analyse.
Filtertype - type filter: 'pass', 'block', 'both'
passdomain– begin and end frequency pass filter.
Blockdomain - begin and end frequency block filter.
newlabel – name of new entry in Series() or Comp() list.
time – start time, end time of timeseries part to filter. Default: whole series.
Output: tide – tide struct with new entry in Series() or Comp() list.

ExportSeries
Purpose: Export series data to file in sds2obs or sds2mat format.
Call: ExportSeries(tide, label, filename, format);
Input: tide – tide struct .
filename – file specification of result file.
label – name of entry in Series() to export.
format – format of datafile ('SDS2MAT', 'OBS2SDS' or 'SIMINP').

ExportComponents
Purpose: Export components to file in sds2obs or sds2mat format.
Call: ExportComponents(tide, label, filename, format);
Input: tide – tide struct .
filename – file specification of result file.
label – name of entry in Comp() to export.

format – format of datafile ('SDS2MAT','OBS2SDS' or 'SIMINP').

AddElements

Purpose:	Add operation on two data elements (Series or components). Result is stored as new entry in tide struct.	

Call:	tide = AddElements(tide, labelA, labelB, newlabel, domain);
	Or:
	tide = AddElements(tide, labelA, labelB, newlabel);	

Input:	tide	- tide struct
	labelA	- name of entry input element A in Series() or Comp().
	labelB	- name of entry input element B in Series() or Comp().
	newlabel	- name of entry result A+B in Series() or Comp().
	domain	- domain specification (time or frequencies). Default: whole interval.
Output:	tide	- tide struct with new entry in Series() or Comp().

SubtractElements

Purpose:	Subtract operation on two data elements (Series or components). Result is stored as new entry in tide struct.	

Call:	tide = SubtractElements(tide, labelA, labelB, newlabel, domain);
	or:
	tide = SubtractElements(tide, labelA, labelB, newlabel);	

Input:	tide	- tide struct
	labelA	- name of entry input element A in Series() or Comp().
	labelB	- name of entry input element B in Series() or Comp().
	newlabel	- name of entry result A-B in Series() or Comp().
	domain	- domain specification (time or frequencies). Default: whole interval.
Output:	tide	- tide struct with new entry in Series() or Comp().

RemoveElement

Purpose:	Remove data element from memory.	
Call:	tide = RemoveElement(tide, label);	
Input:	tide	- tide struct
	label	- name of entry element in Series() or Comp() to remove.
Output:	tide	- tide struct with specified entry in Series() or Comp() re moved.

SetTimeLabels

Purpose:	Define and set ticks and time labels for x-axis of series plot	
Call:	SetTimeLabels(h_axis, day_incr, hour_incr, nTicks); or: SetTimeLabels(h_axis);	
Input:	h_axis	- handle of axis with series plot
	day_incr	- increment for date labels in days. Default: automatic scaling.
	hour_incr	- increment for hour labels in hours. Default: automatic scaling.
	nTicks	- number of ticks between two labels

ShowTide

Purpose:	plot series, components or spectra.
Call:	ShowTide(tide, label, optstring, domain, h_axis); or: ShowTide(tide, label, optstring, domain);

ShowTide(tide, label, optstring);

Input:	tide	- tide struct.
	optstring	- options for color and linestyle.
	domain	- domain specification (time or frequencies)
	h_axis	- handle of axis; Optional; default: new axis will be created.

2.2 Examples

An example of usage of the modules for a simple tidal analysis is given below. In this example the following operations are performed:

- The tide struct is initialized.
- A series of observations is imported from a file 'hvh-1988', in obs2sds format, and is stored in the tide struct with label 'hvh-1988'.
- Part of the series, from 18 january 1998 to 02 august 1998, is analysed for tidal components with a predefined set of tidal constituents for a monthly analysis. The resulting components are stored with label 'comp-hvh'.
- With the generated set of components a prediction is made for the period of 18 january 1998 until 02 august 1998 with an increment of 10 minutes and the result is stored in the tide struct with label 'pred-hvh'.
- The predicted series is subtracted from the original series and the result is stored with label 'res-hvh'.
- From the original series also a fourier analysis is made and stored as 'comp-four-hvh'.
- The fourier components are transformed by an inverse fourier transformation to a new series with label 'hvh-four'.
- The prediction based on the inverse fourier transformation is subtracted from the original series and stored as 'res-four-hvh'.
- Finally the series 'res-hvh', the components 'comp-hvh' and the spectra 'res-four-hvh' are visualised.

The resulting script reads:

```
tide = InitializeTide;
tide = ImportSeries(tide,'hvh1988.dat','HOEKVHLD','OBS2SDS','hvh-1988');
tide = Series2Components(tide,'hvh-1988','month','comp-hvh',{'18 jan 1998 00:00:00' '02 aug 1998
        00:00:00'},0,1,{'P1'});
tide = Component2Series(tide,'comp-hvh','01 jan 1998',{'18 jan 1998 00:00:00' '02 aug 1998
        00:00:00'}, 10.,'pred-hvh');
tide = SubtractElements(tide,'hvh-1988','pred-hvh','res-hvh',{'18 jan 1998 00:00:00' '02 aug 1998
        00:00:00'});
tide = Series2Fourier(tide, 'hvh-1988','comp-four-hvh',{'18 jan 1998 00:00:00' '02 aug 1998
        00:00:00'});
tide = Fourier2Series(tide, 'comp-four-hvh', 'hvh-four');
tide = SubtractElements(tide,'hvh-1988','hvh-four','res-four-hvh',{'18 jan 1998 00:00:00' '31 jul 1998
        00:00:00'});
ShowTide(tide, 'res-hvh', 'g-');
ShowTide(tide, 'comp-hvh', 'r-');
ShowTide(tide, 'res-four-hvh', 'b-');
```

3 User-interface

The user-interface for the model consists of a small main window (see figure 3.1) with a menu bar with the items

- File
- Operations
- Tools
- Help

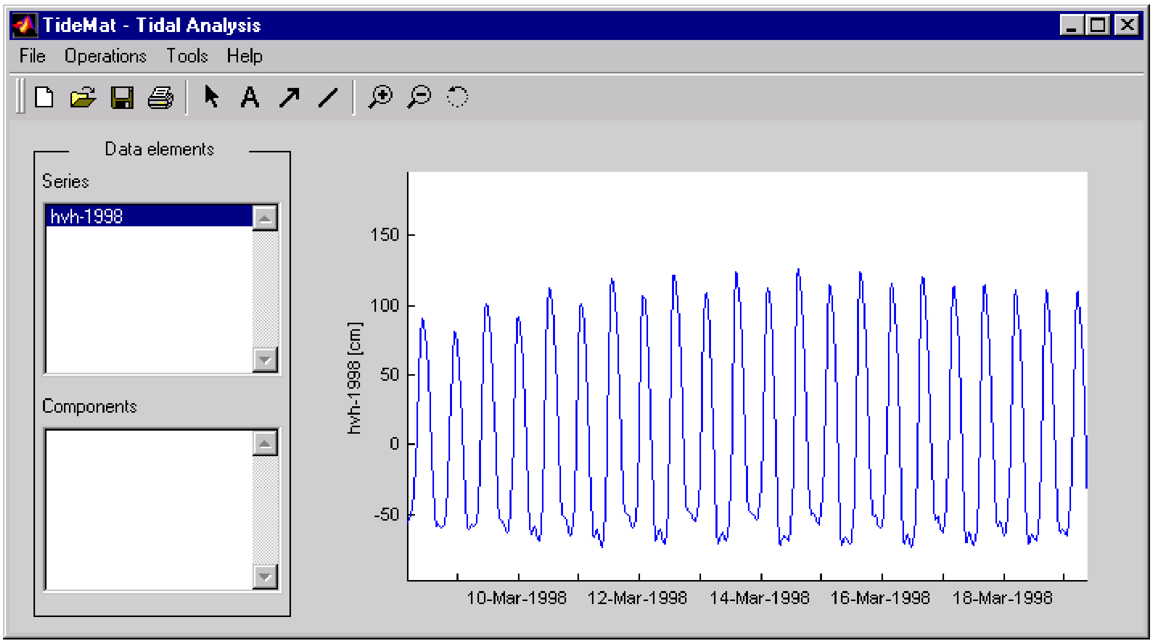

Figure 3.1 Main window of GUI

The listboxes in the main window shows the labels of the available objects, series or components, in the tide struct. Selecting an item in one of the listboxes will result in visualising the item in the remaining part of the window. Depending on the type of object selected, a series plot, a plot of amplitudes and phases of components or a spectrum plot will be presented. The toolbar can be used for several graphical manipulations like zoom in and zoom out.

3.1 File Organisation

The File menu consists of the menu items Import Series and Import Components for importing series or components from file. The items Export Series and Export Components gives the possibility of exporting a series or set of components to a file. The supported file formats are the Simona Obs2Sds, Sds2Mat format and Simona input file format (for specifying boundaries). The menu items Print and Page set-up facilitate printing of the current drawing in the main window.

3.1.1 Import

Series and components can be imported by respectively the Import Series and the Import Components menu item. The dialogue (see figure 3.2) looks the same for both operations. Three different input formats are supported. These formats are:

- obs2sds
- sds2mat
- siminp; Tidemat cannot process whole siminp files but can read the TIMESERIES section (Import Series) and HARMONIC (Import Components) section of the FORCINGS/BOUNDARY

The datafile can be defined directly in the edit field, but can also be selected by a standard file selection dialogue, which is activated when the Browse button is clicked. An Obs2Sds file can contain series or components for several stations and the siminp format for several points (e.g. P12). The desired station/point name must be specified in the "Station name" corresponding edit field. By default the first station/point is read from the file. When series or components are imported from a Sds2Mat, also the runid_constants.mat file is required in the same directory as the datafile. The label which is used to store the data in the tide struct must be given in the edit label field. The siminp-file does not contain a refence date. Therefore the user must enter this date for this type of input. The reference date corresponds to the simulation date of the input file for series. The reference date for components is in general 1-jan-1900 (which is already by default filled-in). For input in siminp format it is also possible to import all points/stations at once by selecting "All stations". Tidemat will generate labels for all the points/stations in that case.

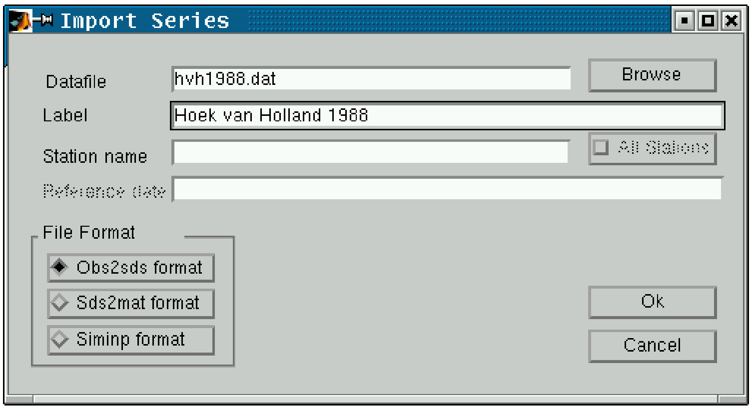

Figure 3.2 Import Series dialogue

3.1.2 Export

Series and component objects which are in the tide struct can be exported to file by means of respectively the Export Series and Export Components menu items. The corresponding dialogues (see figure 3.3) presents a list of available series or components. Fileformat and file specification must be defined and when the Ok button is clicked, the selected series or set of components are written. For the moment it is not possible to export Fourier components.

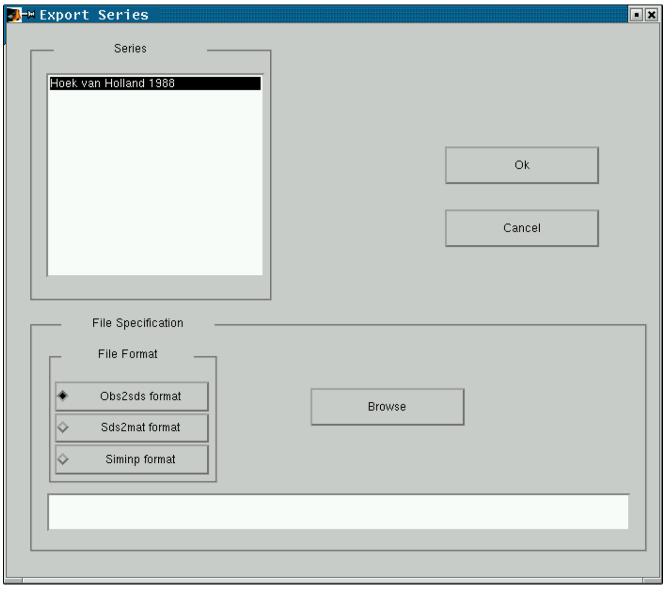

Figure 3.3 Export Series dialogue

3.2 Operations

The Operations menu has the following items:

- Series->Components - Find amplitudes and phases of astronomical components.
- Components->Series -

Make a prediction based on a set of astronomical components.

- Series->Fourier - Compute fourier components.
- Fourier->Series - Transform fourier components to series.
- Filter - Filter operation for series or components.
- Add/Subtract/Remove - Add/subtract two objects or remove an object from memory.

3.2.1 Series->Components

Series can be analysed for astronomical components by means of the 'Series->Components' dialogue (see figure 3.4). This dialogue shows a number of sets of predefined constituents which can be used for analysing the selected series. The set names and the corresponding set of constituents are listed in Appendix B of this report. The user can also define his own set of constituents for analysing the series. In that case he has to select the 'other' option and specify the names of the desired constituents, as listed in Appendix A, in the edit box (one at a line). The edit fields for 'Start time' and 'End time' can be used to define a specific interval of the selected series. The format for specifying date/time is 'dd-mon-yyyy hh:mm:ss'. The controls in the 'Coupled Components' area specify the constituents which will be coupled in the analysis because their frequencies are too close together for the analysed series. After analysing, the coupled components are split up by fixed amplitude relations. The amplitude relations for these constituents are presented in Appendix C of this report. The 'Yes' or 'No' control in the 'Transform to 1-1-1900' area specifies whether or not the components are transformed to a reference date of 1 January 1900. Computation of nodal factors is included if transformation to 1 January 1900 is selected. The edit field for 'Component label' is required to specify a label which is used to store the computed components in the tide struct.

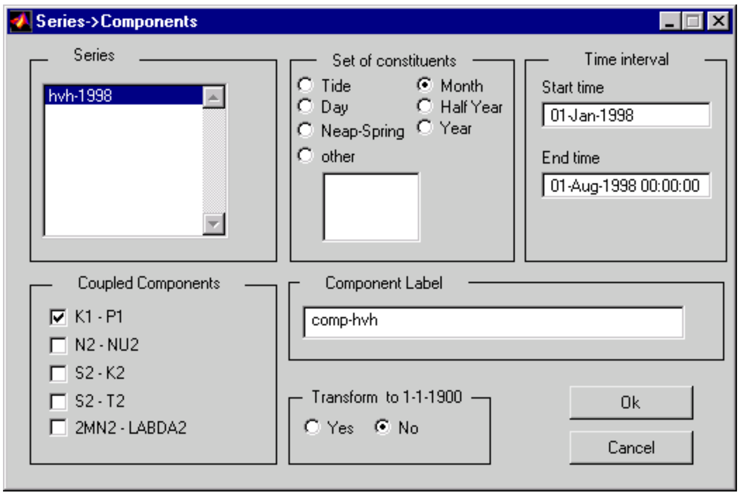

Figure 3.4 Series->Components dialogue

3.2.2 Component->Series

A prediction for arbitrary time interval out of a set of components can be made by the 'Component-

‹ with all available
ea the reference
ıd of the time inter-
ed, must be given (in
ontrols in the 'Cou-
frequencies are
The amplitude rela-
field for 'Series la-
tide struct.

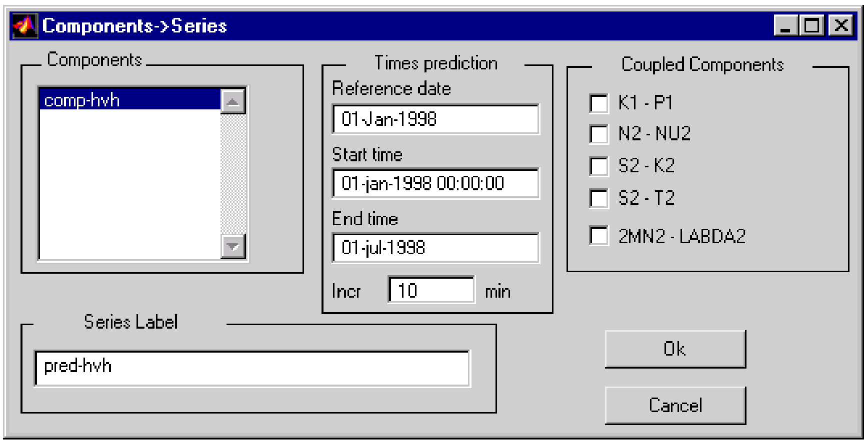

Figure 3.5 Components->Series dialogue

3.2.3 Series->Fourier

Computing the Fourier components of a series can be done with the 'Series->Fourier' operation (see figure 3.6). The dialogue shows a list of labels of the available series from which the desired one must be selected. The fields for 'Start time' and 'End time' specify which time interval of the selected series must be used for the analyse. The format for specifying date/time is 'dd-mon-yyyy hh:mm:ss'. The edit field for 'Component label' is required to specify a label which is used to store the computed components in the tide struct.

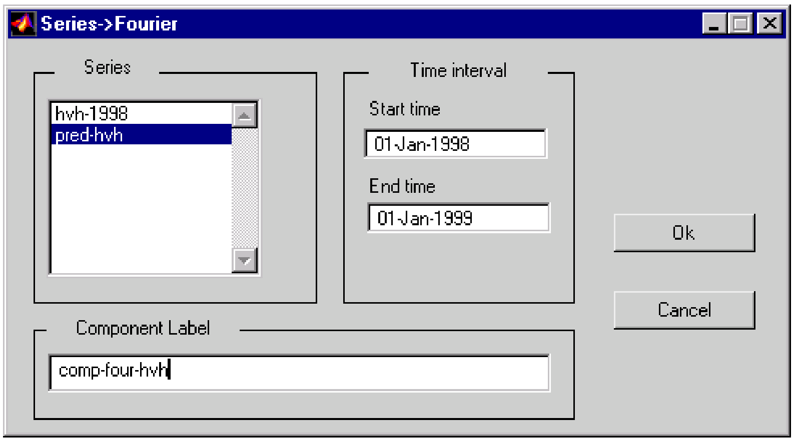

Figure 3.6 Series->Fourier dialogue

3.2.4 Fourier->Series

The inverse Fourier transformation can be performed with the 'Fourier->Series' operation (see figure 3.7). The dialogue shows a list of labels of available sets of Fourier components, from which the desired one must be selected. The edit field for 'New series label' is required to specify a label which is used to store the resulting series in the tide struct

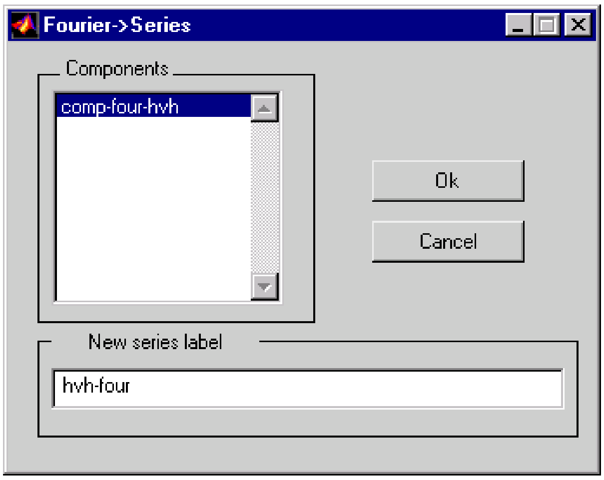

Figure 3.7 Fourier->Series dialogue

3.2.5 Filter operation

Both series and components can be filtered by removing specified frequency components. Filtering of series is done by transforming the series into fourier components, filter the

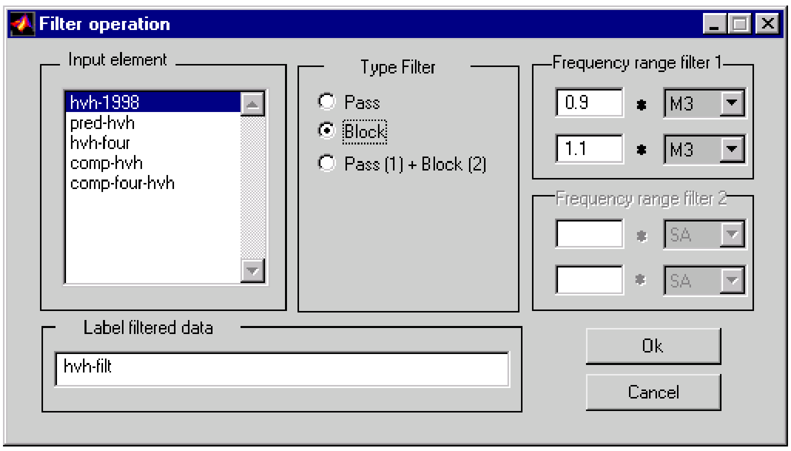

Figure 3.8 Filter dialogue

components and perform an inverse fourier transformation for the filtered components. The dialogue for the filter operation (see figure 3.8) show a list of available input objects (series and components), from which the desired one must be selected. Three different types of filters can be specified. The 'Pass' filter allows to pass the frequencies within the specified interval for the filter and blocks all frequencies outside this interval. The 'Block' filter blocks the frequencies which are within the specified interval and allows to pass all other frequencies. The third filter is a combination of the 'Pass' and 'Block' filter and it requires to specify and second frequency interval for blocked frequencies within the 'Pass' interval. Begin and end value of the frequency interval(s) are specified by selecting a constituent from the list and specifying a multiplication factor. So in the dialogue of figure 3.8 the interval is started at 0.9 times the frequency of constituent 'M3' and ends at 1.1 times the frequency of 'M3'. The edit field for 'Label filtered data' is required to specify a label which is used to store the resulting object in the tide struct

3.2.6 Add/Subtract/Remove operation

Add and subtract operations can be performed for series and components (see figure 3.9). This dialogue also can be used to remove a series or a set of components from the tide struct and out of memory. The dialogue shows two lists of labels of all available objects to select the elements (A and B) for the operation. The fields for 'Start time' and 'End time' specify the time interval to use for the operation. If the input elements are components instead of a time interval, a frequency interval can be specified. The format for specifying date/time is 'dd-mon-yyyy hh:mm:ss'. The edit field for 'New Label' is required to specify a label which is used to store the resulting object in the tide struct. When the 'Remove' operation is selected, only element (A) needs to be selected and all other controls are disabled.

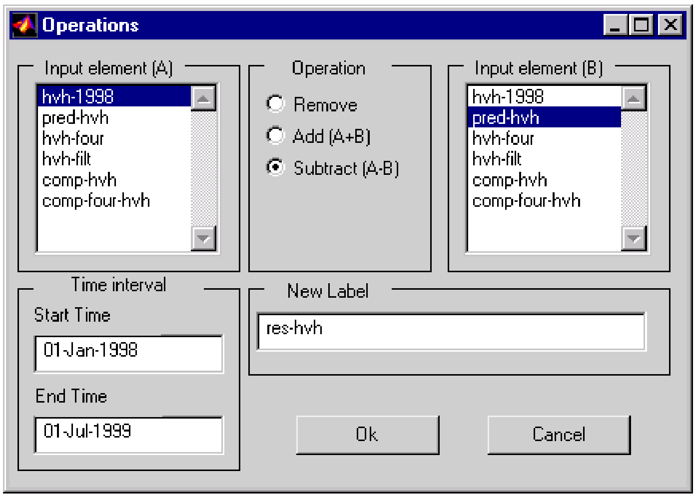

Figure 3.9 Operations dialogue

3.3 Tools

3.3.1 Options

The 'Tools' menu has the submenu 'Options', to specify a few plot options to control the current plot in the main window. In the options dialogue (see figure 3.10) the label(s) for the y-axis of the series plot, component plot (amplitudes and phases) or spectra plot can be defined. By default the time labels at the x-axis of a series plot are computed automatic. In the options dialogue however 'Manual' scaling can be selected and the number of day and time labels can be controlled. All defined options are only active for the current plot in the main window.

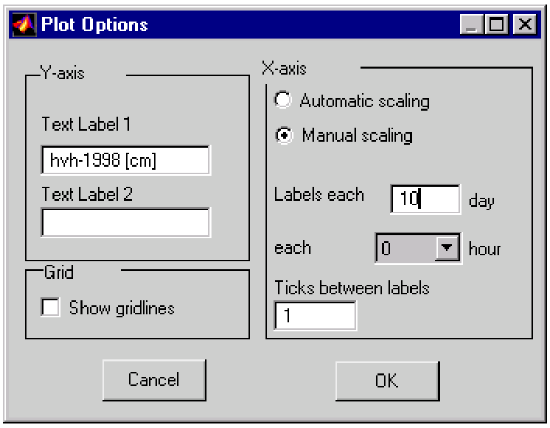

Figure 3.10 Options dialogue

3.3.2 Macro

Some (series of) operations with specific options are to be applied multiple times. Macros can be used to simplify this task. Once a macro is created it can be applied for on different input and saved for later use. The Macro dialog box appears (see figure 3.11) when the user selects the Macro option from the Tools menu. The Macro dialog consists of the following buttons:

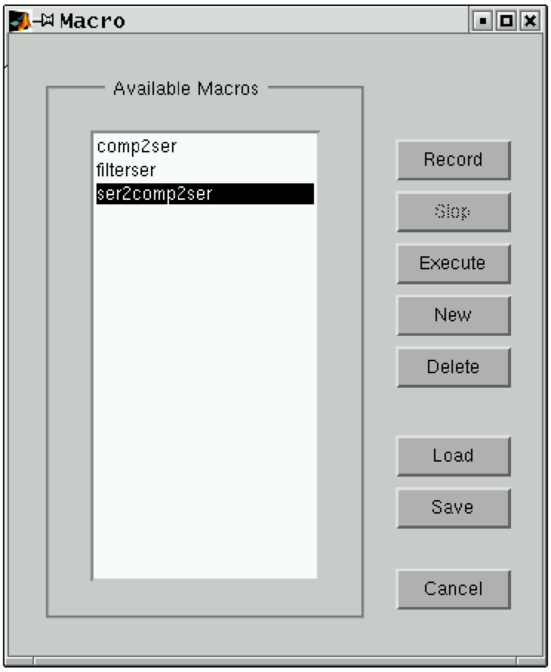

Figure 3.11 Macro dialogue

- Record: start recording the selected macro. The information in the selected macro is deleted!
- Stop: Stop recording
- Execute: apply the selected macro on the selected input (series and or component) in the main window.
- New: create a new, empty macro
- Delete: Delete the selected macro
- Load: import a macro from file
- Save: save the selected macro to file
- Cancel: close window, note: this will not undo any action.

The filename of a macro is not necessarily the same as the name of the macro. If a macro is loaded from file that is already present it will be renamed.
Only actions from the Operations menu item are stored in the macro in this version of tidemat.

Appendix A

List of constituents

Name	Frequency deg/hour	Name	Frequency deg/hour
SA	0.0410	MNS2	27.4238
SSA	0.0821	2ML2S2	27.4966
MSM	0.4715	2MS2K2	27.8039
MM	0.5443	NLK2	27.8860
MSF	1.0158	2N2	27.8953
SM	1.0158	MU2	27.9682
MF	1.0980	2MS2	27.9682
SNU2	1.4874	SNK2	28.3575
SN	1.5602	N2	28.4397
MFM	1.6424	NU2	28.5125
2SM	2.0317	2KN2S2	28.6040
2SMN	2.5761	OP2	28.9019
2Q1	12.8542	MSK2	28.9019
NJ1	12.8542	MPS2	28.9430
SIGMA1	12.9271	M2	28.9840
NUJ1	12.9271	MSP2	29.0251
Q1	13.3986	MKS2	29.0662
RO1	13.4714	M2(KS)2	29.1483
NUK1	13.4714	2SN(MK)2	29.3734
O1	13.9430	LABDA2	29.4556
TAU1	14.0251	2MN2	29.5284
MP1	14.0251	L2	29.5284
M1B	14.4874	L2A	29.5284
M1C	14.4920	L2B	29.5377
M1D	14.4920	2SK2	29.9178
M1A	14.4967	T2	29.9589
M1	14.4967	S2	30.0000
NO1	14.4967	R2	30.0410
CHI1	14.5695	K2	30.0821
LP1	14.5695	MSN2	30.5443
PI1	14.9178	ETA2	30.6265
TK1	14.9178	KJ2	30.6265
P1	14.9589	MKN2	30.6265
S1	14.9999	2KM(SN)2	30.7086
K1	15.0410	2SM2	31.0159
PSI1	15.0821	SKM2	31.0980
RP1	15.0821	2SNU2	31.4874
FI1	15.1231	3(SM)N2	31.4874
KP1	15.1231	2SN2	31.5602
THETA1	15.5125	SKN2	31.6423
LABDAO1	15.5125	MQ3	42.3826
J1	15.5854	NO3	42.3826
2PO1	15.9748	MO3	42.9271
SO1	16.0569	2MK3	42.9271
OO1	16.1390	2MP3	43.0093
KQ1	16.6834	M3	43.4761
3MKS2	26.8701	SO3	43.9429
3MS2	26.9523	MK3	44.0251
OQ2	27.3417	2MQ3	44.5696
MNK2	27.3417	SP3	44.9588
SK3	45.0410	MKNU6	87.5786

Name	Frequency deg/hour	Name	Frequency deg/hour
K3	45.1231	2(MS)K6	87.8859
2SO3	46.0568	2MS6	87.9681
4MS4	55.9362	2MK6	88.0503
2MNS4	56.4079	2SN6	88.4398
3MK4	56.8703	3MSN6	88.5126
MNLK4	56.8703	MKL6	88.5948
3MS4	56.9524	2SM6	88.9840
MSNK4	57.3416	MSK6	89.0661
MN4	57.4237	S6	89.9998
2MLS4	57.4966	2MNO7	100.3509
2MSK4	57.8861	2NMK7	100.9047
M4	57.9683	M7	101.4489
2MKS4	58.0504	2MSO7	101.9113
SN4	58.4396	MSKO7	103.0093
3MN4	58.5125	2(MN)8	114.8475
2SMK4	58.9020	3MN8	115.3920
MS4	58.9841	3MNKS8	115.4742
MK4	59.0663	M8	115.9362
2SNM4	59.4554	2MSN8	116.4079
2MSN4	59.5283	2MNK8	116.4901
SL4	59.5283	3MS8	116.9521
S4	60.0000	3MK8	117.0343
SK4	60.0821	2SNM8	117.4238
2SMN4	60.5442	MSNK8	117.5059
3SM4	61.0158	2(MS)8	117.9683
2SKM4	61.0980	2MSK8	118.0505
MNO5	71.3669	3SM8	118.9841
3MK5	71.9111	2SMK8	119.0663
3MP5	71.9932	S8	120.0000
M5	72.4649	2(MN)K9	129.8887
MNK5	72.4649	3MNK9	130.4332
2MP5	72.9269	4MK9	130.9774
3MO5	73.0091	3MSK9	131.9933
MSK5	74.0250	4MN10	144.3760
3KM5	74.1071	M10	144.9206
2(MN)S6	84.8477	3MSN10	145.3919
3MNS6	85.3919	4MS10	145.9364
2NM6	85.8635	2(MS)N10	146.4077
4MS6	85.9364	2MNSK10	146.4899
2MSNK6	86.3259	3M2S10	146.9523
2MN6	86.4077	4MSK11	160.9776
2MNU6	86.4806	M12	173.9046
3MSK6	86.8701	4MSN12	174.3762
M6	86.9523	5MS12	174.9204
MSN6	87.4239	3MNKS12	175.4742
4MN6	87.4968	4M2S12	175.9363
MNK6	87.5061	A0	0.0000

Appendix B

List of predefined sets of constituents

Set	Constituensts					
Tide	A0	M2	M4	M6	M8	M10	M12
Day	A0	M1	M2	M3	M4	M5	M6
	M7	M8	M10	M12			
Neap-spring	A0	O1	K1	MU2	M2	S2	2SM2
	3MS4	M4	MS4	M6	2MS6	M8	3MS8
	4MS10						
Month	A0	Q1	O1	K1	3MS2	MNS2	MU2
	M2	2MN2	S2	2SM2	3MS4	MN4	M4
	MS4	2MN6	M6	2MS6	M8	3MS8	4MS10
Half Year	A0	SA	SM	Q1	O1	M1C	P1
	K1	3MKS2	3MS2	OQ2	MNS2	2ML2S2	NLK2
	MU2	N2	NU2	M2	LABDA2	2MN2	S2
	K2	MSN2	2SM2	SKM2	NO3	2MK3	2MP3
	SO3	MK3	SK3	4MS4	2MNS4	3MS4	MN4
	2MLS4	2MSK4	M4	3MN4	MS4	MK4	2MSN4
	S4	MNO5	3MK5	2MP5	3MO5	MSK5	3KM5
	3MNS6	2NM6	4MS6	2MN6	2MNU6	3MSK6	M6
	MSN6	MKNU6	2MS6	2MK6	3MSN6	2SM6	MSK6
	2MNO7	M7	2MSO7	2(MN)8	3MN8	M8	2MSN8
	2MNK8	3MS8	3MK8	2(MS)8	2MSK8	3MNK9	4MK9
	3MSK9	4MN10	M10	3MSN10	4MS10	2(MS)N10	3M2S10
	4MSK11	M12	4MSN12	5MS12	4M2S12		
Year	A0	SA	SM	Q1	O1	M1C	P1
	S1	K1	3MKS2	3MS2	OQ2	MNS2	2ML2S2
	NLK2	MU2	N2	NU2	MSK2	MPS2	M2
	MSP2	MKS2	LABDA2	2MN2	T2	S2	K2
	MSN2	2SM2	SKM2	NO3	2MK3	2MP3	SO3
	MK3	SK3	4MS4	2MNS4	3MS4	MN4	2MLS4
	2MSK4	M4	3MN4	MS4	MK4	2MSN4	S4
	MNO5	3MK5	2MP5	3MO5	MSK5	3KM5	3MNS6
	2NM6	4MS6	2MN6	2MNU6	3MSK6	M6	MSN6
	MKNU6	2MS6	2MK6	3MSN6	2SM6	MSK6	2MNO7
	M7	2MSO7	2(MN)8	3MN8	M8	2MSN8	2MNK8
	3MS8	3MK8	2(MS)8	2MSK8	3MNK9	4MK9	3MSK9
	4MN10	M10	3MSN10	4MS10	2(MS)N10	3M2S10	4MSK11
	M12	4MSN12	5MS12	4M2S12			

Appendix C

Amplitude relations of coupled components

Component	Amplitude relation
P1	0.328 * K1
NU2	0.194 * N2
LABDA2	0.005 * M2
2MN2	0.029 * M2
T2	0.059 * S2
K2	0.284 * S2

Appendix D

Description of tide struct

Datastructure

The datastructure of the program consistswill of the next elements:

- Tide - overall structure covering all data elements
- Tdata - structure with information on all harmonic constituents
- Series - list with structs of timeseries data
- Comp - list with structs of fourier components of analysed data

Each of the elements mentioned above is of the type struct (Matlab data-type). These structs consists of other structs and/or a number of basic data items, like integers, reals or strings. Below the contents of each struct and their hierarchy is shown.

Tide –

Tdata	- struct with harmonic constituents
Series()	- list with structs of timeseries data
Comp()	- list with structs of fourier components
Options	- struct with plot options
Macros	- struct containing the macros

Tdata –

Constants	- struct with several constants
names{1:196}	- names of harmonic constituents
bnames(1:196)	- name of coupled component
freq{1:196}	- frequencies of harmonic constituents [rad/min]
Sets	- sets of predefined selections of tide components
amplfac(1:196)	– amplitude factors for coupled component

Series –

label	- unique name
type	- type keyword 'analyse' / 'residu' / 'predict' /
Station	- struct with station data
Timedef	- struct with information time data
Timeform	- struct with format specification time data
Values	- struct with general values
Data	- struct with data

Comp –

label	- unique name for components
type	- type keyword 'ASTRO' : harmonic components 'FOURIER' : fourier components
Timedef	- struct with information time data
Timeform	- struct with format specification time data
Station	- struct with station data
Values	- struct with general values
names{}	- names of analysed harmonic constituents
freqs()	- frequencies of analysed harmonic constituents [rad/hour]
amplitudes()	- amplitudes of Fourier components
phases()	- phases of Fourier components

Macros		
	Record	Record vlag: 0->record off otherwise index in Macro() that is recorded
	Play	Play flag similar to Record
	StepNr	Step of macro that is last recorded/played
	Macro()	array of macro structs
Constants –		
	second	- unit for second [minutes]
	minute	- unit for minute [minutes]
	hour	- unit for hour [minutes]
	day	- unit for day [minutes]
	month	- unit for month (30 days) [minutes]
	year	- unit for year (365 days) [minutes]
	tide	- unit for tide (12.30 hours) [minutes]
	radian	- unit for radians (1) [rad]
	cycle	- unit for cycle (2¶ * radian) [rad]
	dn_smooth	- number of samples to use for smoothed transfer begin and end of series
	M1-cycle	- length of M1 component
Sets –		
	tide	- harmonic constituents for 12.3 hour analysis
	neap_spring	- harmonic constituents for neap-spring analysis
	month	- harmonic constituents for 30 days analysis
	half_year	- harmonic constituents for half year analysis
	year	- harmonic constituents for one year analysis
Station		
	name	- name of station
	Coordinates	- struct with info co-ordinates
Coordinates		
	system	- name of used co-ordinate system 'ED50', 'RDV', 'INDEX', 'UTM'
	zone	- utm zone
	zpos	- z-position of station w.r.t. z-reference in mtr.
	zref	- z-reference level for z-position. 'NAP', 'Waterlevel', 'Bottom'
	bottom	- bottom level w.r.t. NAP
	Location	- struct with coordinates of location
Timedef		
	tzone	- Time zone 'MET', 'GMT', 'UTC'
	offset	- offset in minutes.
	refdate	- string with reference date for timeseries
Timeform		
	serieform	- form of series data 'REGULAR', 'IRREGULAR'
	Timeframe	- struct with time frame for regular data.
	dateformat	- string with format of date/time data.

Values

	type	- type of data: 'waterlevel', 'current', etc
	dummy	- value of dummy data
	unit	- unit of data 'cm', 'dm', 'm'
	cvfactor	- conversion factor

Data

	times()	- array with time levels related to refdate [minutes]
	zvalues()	- values of data

Timeframe

	start	- start time in minutes w.r.t. refdate
	end	- end time in minutes w.r.t. refdate
	incr	- time increment in minutes

Location

	Unit	- 'deg', 'm', '-'
	Xp	- x-coordinate
	Yp	- y-coordinate

Macro

	Name	Name of macro
	Step{}	Cell array of Step structs each containing one operation/step of the macro

Step

	kind	Name of the operation
	Input	struct containing information on input arguments of step
	Output	struct containing information on output arguments of step
	Eval	String using optional attributes of Step struct that is used to evaluate step in matlab
	<optionals>	optional attributes needed to evaluate step. Depending on operation

Input

	Name{}	Name of input arguments in <optionals>
	Datatype{}	Type of input arguments ("Series" or "Comp").