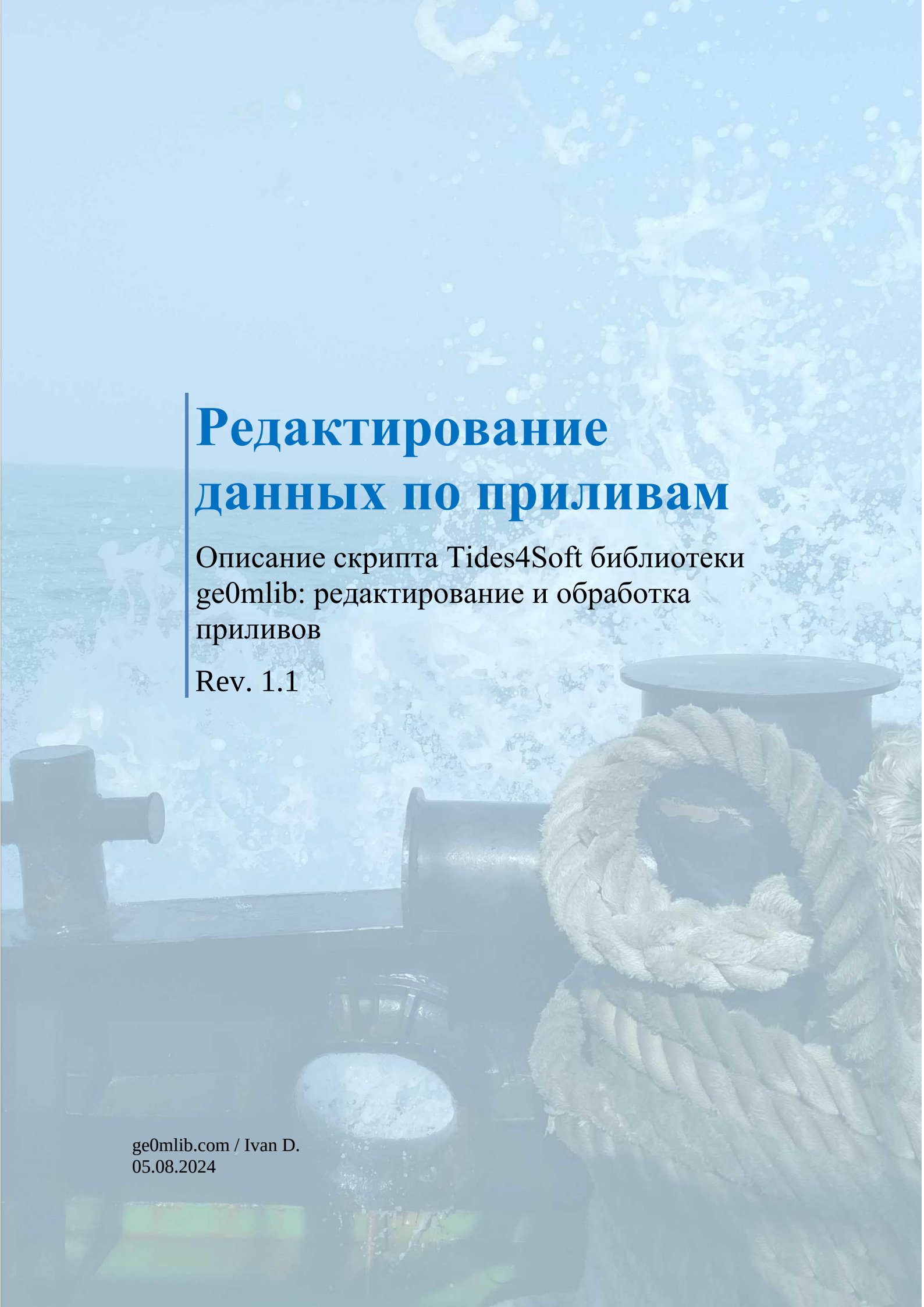




Редактирование данных по приливам

Описание скрипта Tides4Soft библиотеки
ge0mlib: редактирование и обработка
приливов

Rev. 1.1

Содержание

Введение	3
1. Форматы данных.....	4
2. Команды скрипта	6
Приложение 1. Код скрипта Tides4Soft	11

История изменений

Версия	Дата	Описание изменений
1.0	2022-08-14	Документ создан // Иван Д.
1.1	2024-08-05	Документ пересмотрен // Иван Д.

Список рисунков

Figure 0.1 Скриншоты окон MatLab, с примерами сессий команд скрипту Tides4Soft.....	3
Figure 2.1 Входные файлы.....	6
Figure 2.2 График загруженных данных по приливам	6
Figure 2.3 График «разряженных» данных	7
Figure 2.4 Окно MatLab для интерактивного удаления вылетов	8
Figure 2.5 Окно с инструментом, удаляющим измерения в выделенной прямоугольной области	8
Figure 2.6 Окно с графиками, отображающими результаты редактирования	9
Figure 2.7 Исходные данные (синие точки) и обработанные данные, интерполированные к интервалу 1 минута (красные точки)	10
Figure 2.8 Содержимое папки с файлами формата Silas	10

Введение

Скрипт Tides4Soft написан для редактирования текстовых файлов с данными приливов (часто такое редактирование выполняется в Excel, что отнимает у обработчика много времени). Скрипт реализует следующие шаги обработки:

- загрузка всех файлов с приливами из определенной папки в MatLab и объединение их в единый массив данных;
- децимация массива для быстро сглаживания данных по приливам;
- интерактивное удаление выбросов при помощи инструмента gMapPickHandleNan,
- сглаживание, при заданной ширине скользящего окна,
- интерполяция удаленных на предыдущем шаге значений,
- интерполяция данных к шагу в 1 минуту,
- сохранение обработанных данных о приливах в текстовом файле.

Для запуска скрипта необходимо установить программное обеспечение MatLab 2018a (или более позднюю версию) и библиотеку geOmlib.

Результаты обработки сохраняются в форматах, читаемых программным обеспечением DugInsight или Silas (формат tdx). На [Figure 0.1](#) показаны скриншоты окон MatLab с примерами сессий команд скрипту Tides4Soft (для редактирования текстовых файлов с данными приливов).

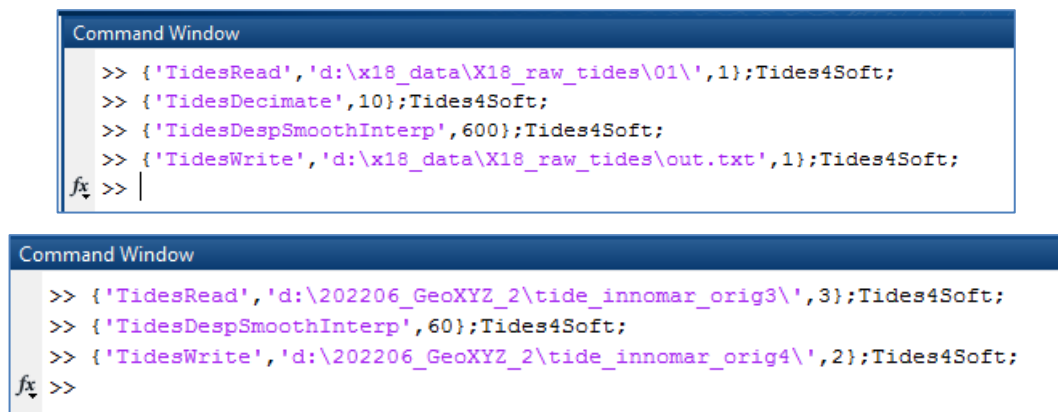


Figure 0.1 Скриншоты окон MatLab, с примерами сессий команд скрипту Tides4Soft

1. Форматы данных

В коде скрипта реализованы три формата для исходных данных:

- 1) данные получаемые с USV, без записи даты регистрации внутри текстового файла;
- 2) данные получаемых с USV, с записью даты регистрации внутри текстового файла;
- 3) данные в формате, наиболее часто используемом в приливных станциях.

Пример формата «1» для исходных данных (для USV, без записи даты регистрации)

```
#<Time> HHmmss.zzz, Latitude WGS-84 - COG - Latitude WGS-84, Longitude WGS-84 - COG - Longitude WGS-84, Height WGS-84 - COG - Height WGS-84, Easting - COG - Easting, Northing - COG - Northing, Height - COG - Height
```

```
161248.237,5520.17000858,00134.00310710,54.775,580051.32,6100123.86,8.44
```

```
161249.001,5520.17000347,00134.00310369,54.777,580051.33,6100123.85,8.44
```

Последний столбец содержит значение приливов (чаще всего – в сантиметрах).

Дата регистрации для формата «1», считывается из имени файла, в котором содержатся данные; при этом символы, содержащие дату, «вырезаются» из имени файла с помощью оператора fName(end-16:end-9). Пример имён файлов:

[X18_Tide_20220715_0000.log](#); [X18_Tide_20220716_0000.log](#)

Пример формата «2» для исходных данных (для USV, с датой регистрации)

```
#<Time> HHmmss.zzz,<Date> dd,<Date> MM,<Date> yyyy, Latitude WGS-84 - COG - Latitude WGS-84, Longitude WGS-84 - COG - Longitude WGS-84, Height WGS-84 - COG - Height WGS-84, Easting - COG - Easting, Northing - COG - Northing, Height - COG - Height
```

```
000001.002,19,07,2022,5516.94135841,00117.68878421,46.600,608305.37,6127537.33,0.87
```

```
000002.002,19,07,2022,5516.94081535,00117.68899275,46.623,608305.17,6127536.32,0.89
```

Пример формата «3» для исходных данных

```
23:27:38,2.39
```

```
23:28:08,2.60
```

Дата регистрации для формата «3», считывается из имени файла, в котором содержатся данные; при этом символы, содержащие дату, «вырезаются» из имени файла с помощью оператора fName(end-11:end-4). Пример имён файлов:

[tide_innomar_20220512.txt](#); [tide_innomar_20220513.txt](#)

Внимание! Строки, не соответствующие формату, необходимо удалить из входных файлов перед использованием скрипта. Пример строк, не соответствующих формату «3»:

```
13:57:32,
```

```
13:58:02,
```

В коде скрипта реализованы два формата для обработанных данных:

- 1) формат, читаемый программным обеспечением DugInsight;
- 2) формат tdx, читаемый программным обеспечением Silas.

Пример формата «1» для обработанных данных (для DugInsight):

1612:00 15 7 2022 4.74

1613:00 15 7 2022 4.74

Пример формата «2» для обработанных данных (tdx-формат для Silas):

%08:40 1,974

%08:41 2.007

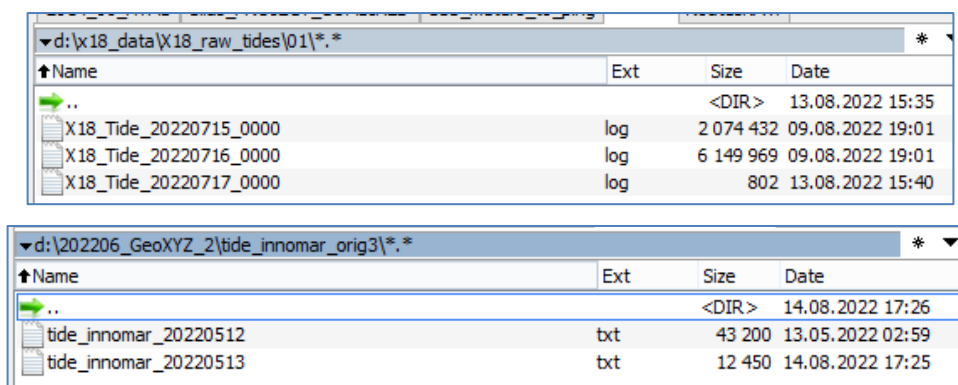
Обработанные данные помещаются в текстовые файлы, созданные отдельно для каждого дня.

Пример имён обработанных файлов:

[Tide_20220512.tdx](#); [Tide_20220513.tdx](#)

2. Команды скрипта

Перед запуском скрипта необходимо поместить файлы с исходными данными в отдельную папку. Пример содержимого папок для формата «1» и формата «3» показан на [Figure 2.1](#).



Name	Ext	Size	Date
..		<DIR>	13.08.2022 15:35
X18_Tide_20220715_0000	log	2 074 432	09.08.2022 19:01
X18_Tide_20220716_0000	log	6 149 969	09.08.2022 19:01
X18_Tide_20220717_0000	log	802	13.08.2022 15:40

Name	Ext	Size	Date
..		<DIR>	14.08.2022 17:26
tide_innomar_20220512	txt	43 200	13.05.2022 02:59
tide_innomar_20220513	txt	12 450	14.08.2022 17:25

[Figure 2.1](#) Входные файлы

Для форматов данных «1» и «3», дата записи считывается скриптом из имени файла. Положение символов, определяющих дату в имени файла, рассчитывается как fName(end-16:end-9) для формата «1» и как fNameN(end-11:end-4) для формата «3».

Ниже описаны команды скрипта, реализующие последовательные шаги обработки.

1) Загрузка файлов с исходными данными выполняется при помощи команды скрипта вида
`>> {'TidesRead','d:\x18_data\X18_raw_tides\01\',1};Tides4Soft;`

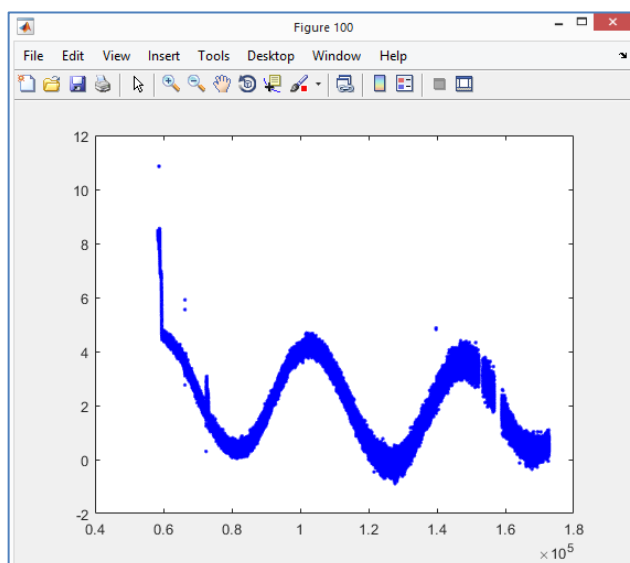
где

'TidesRead' – имя команды скрипта;

'd:\x18_data\X18_raw_tides\01\' – имя папки с исходными данными;

1 – номер формата исходных данных (1, 2 или 3).

После загрузки данных скрипт строит график приливов вдоль временной оси ([Figure 2.2](#)).



[Figure 2.2](#) График загруженных данных по приливам

2) В случае, если необходимо уменьшить объем данных, используется команда скрипта вида

```
>> {'TidesDecimate',10};Tides4Soft;
```

где

'TidesDecimate' – имя команды скрипта;

10 – шаг, с которым сохраняются данные (для приведенного примера команды будет сохранено каждое 10-е значение из ряда приливов).

После выполнения команды скрипт строит график с «разряженными» данными (*Figure 2.3*).

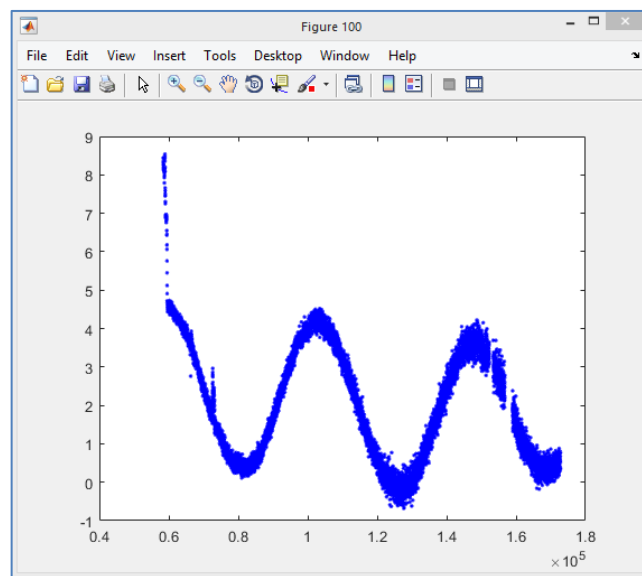


Figure 2.3 График «разряженных» данных

3) Для удаления вылетов, сглаживания и интерполяции удаленных значений, используется команда скрипта вида

```
>> {'TidesDespSmoothInterp',600};Tides4Soft;
```

где

'TidesDespSmoothInterp' – имя команды скрипта;

600 – ширина скользящего окна для сглаживания данных по приливам.

После запуска команды на исполнение открывается графическое окно MatLab для интерактивного удаления вылетов (*Figure 2.4*)

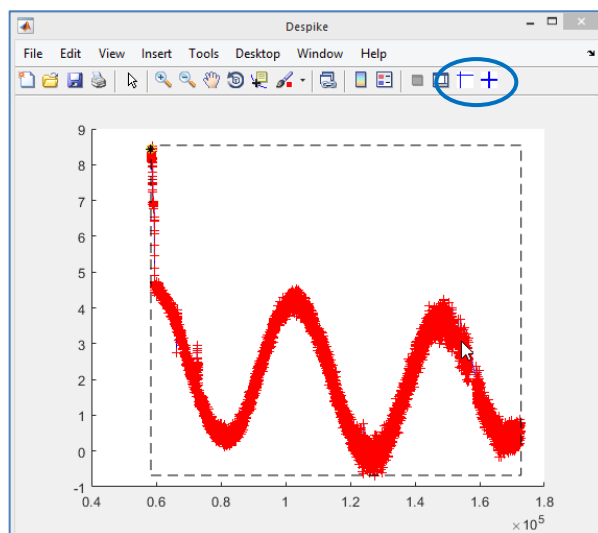


Figure 2.4 Окно MatLab для интерактивного удаления вылетов

Вылеты можно удалять с помощью двух инструментов (иконки обведены синей окружностью на **Figure 2.4**). Работа первого инструмента, удаляющего измерения в выделенной прямоугольной области, показана на **Figure 2.5**.

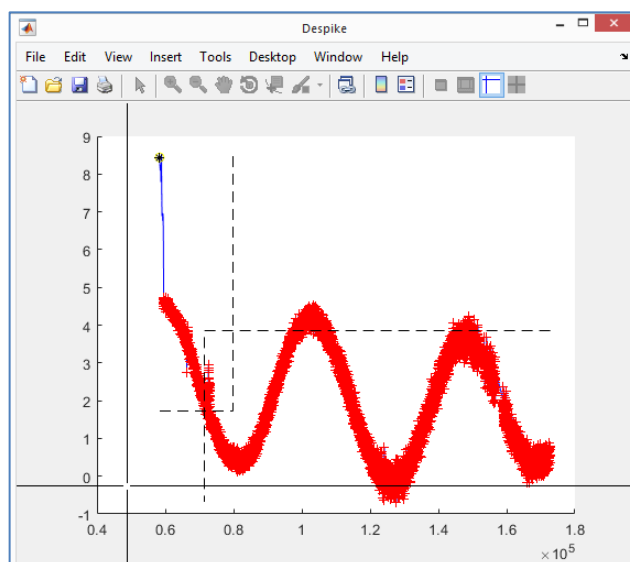


Figure 2.5 Окно с инструментом, удаляющим измерения в выделенной прямоугольной области

При работе с окном удаления выбросов (при помощи выделения прямоугольной области) используются следующие клавиши:

Нажатие на значок – перейти в режим «редактирования» (удаления выбросов);

Левая кнопка мыши – режим редактирования, установить первую границу выделенной области (первая часть прямоугольника или первая точка кривой);

Правая кнопка мыши – режим редактирования, установить вторую границу выделенной области (вторая часть прямоугольника или вторая точка кривой);

Колёсико мыши – режим редактирования, «удалить» точки в выделенной области (значение координаты Y установлено как NaN);

z – режим редактирования, отменить действие;

x – режим редактирования, вернуть отмененное действие;

q – выйти из режима редактирования;

Backspace (или любая клавиша) – закрыть окно удаления выбросов (клавиша нажимается после выхода из режима «редактирования»);

После выхода из режима «редактирования» (нажатие на клавишу «q») можно изменять масштаб графика или выполнять его сдвиг, нажимая на соответствующие иконки (обведены синей окружностью на [Figure 2.6](#)).

После закрытия окна удаления выбросов, в графическом окне MatLab ([Figure 2.6](#)) строятся графики, отображающие результаты редактирования. Окно содержит следующие графики:

Синие точки – исходные данные;

Красная линия – сглаженные данные;

Голубая линия – результат интерполяции сглаженных данных, при котором удаленные в режиме редактирования значения «восстанавливаются» при помощи линейной интерполяции.

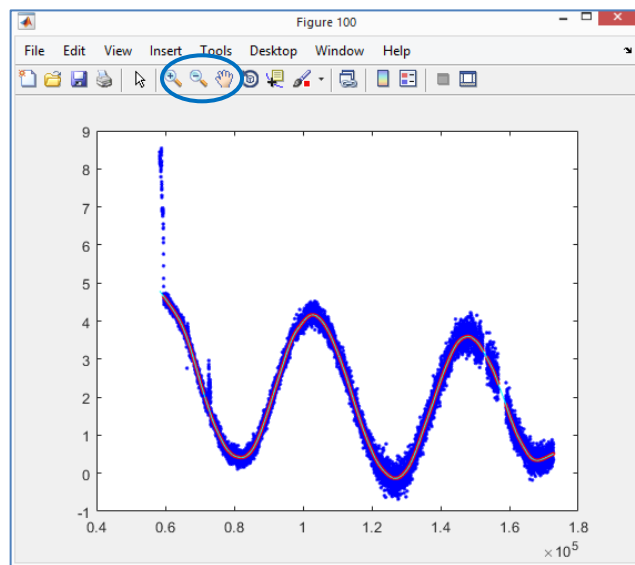


Figure 2.6 Окно с графиками, отображающими результаты редактирования

4) Сохранение результатов обработки в формате DugInsight, выполняется при помощи команды скрипта вида

```
>> {'TidesWrite','d:\x18_data\X18_raw_tides\out.txt',1};Tides4Soft;
```

где

'TidesWrite' – имя команды скрипта;

'd:\x18_data\X18_raw_tides\out.txt' – имя файла для вывода данных;

1 – номер формата, соответствующего DugInsight.

Сохранение результатов обработки в формате tdx (Silas), выполняется при помощи команды скрипта вида

```
>> {'TidesWrite','d:\202206_GeoXYZ_2\tide_innomar_orig4',2};Tides4Soft;
```

где

'TidesWrite' – имя команды скрипта;

'd:\202206_GeoXYZ_2\tide_innomar_orig4\' – имя папки для вывода данных;

2 – номер формата, соответствующего tdx (Silas).

После запуска команды на исполнение, шаг данных по приливам приводится к интервалу в одну минуту, после чего данные сохраняются в файл. После выполнения команды, скрипт строит графики, показанные на **Figure 2.7**:

Синие точки – исходные данные;

Красные точки – обработанные данные с шагом в 1 минуту.

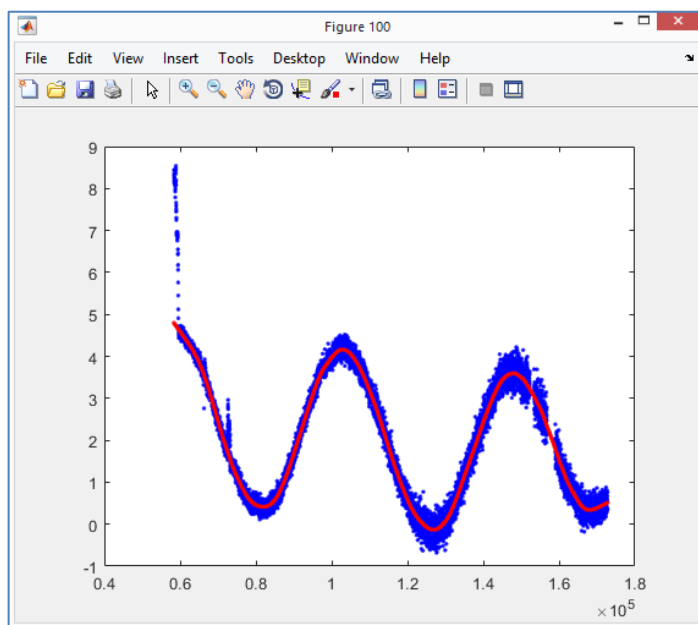
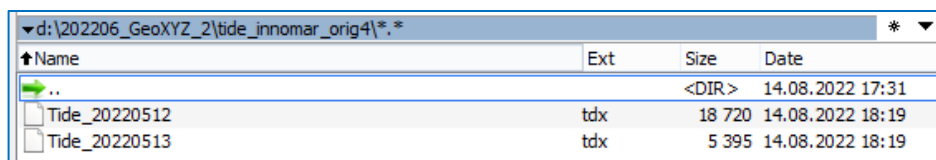


Figure 2.7 Исходные данные (синие точки) и обработанные данные, интерполированные к интервалу 1 минута (красные точки)

Для формата обработанных данных «1» все данные выгружаются в один файл. Для формата обработанных данных «2» (tdx – Silas) файлы создаются для каждого дня отдельно (**Figure 2.8**).



Name	Ext	Size	Date
..		<DIR>	14.08.2022 17:31
Tide_20220512	tdx	18 720	14.08.2022 18:19
Tide_20220513	tdx	5 395	14.08.2022 18:19

Figure 2.8 Содержимое папки с файлами формата Silas

Сохраненные файлы можно использовать для загрузки данных о приливах в программное обеспечение DugInsight или Silas.

Приложение 1. Код скрипта Tides4Soft

```
%script Tides4Soft;
%Edit tides and transform from equipment data format to software data format (DugInsight, Silas)
%==== Input format 1 (USV data format) >>>
%#<Time> HHmmss.zzz, Latitude WGS-84 - COG - Latitude WGS-84, Longitude WGS-84 - COG - Longitude WGS-84, Height WGS-84 - COG - Height WGS-84, Easting - COG - Easting, Northing - COG - Northing, Height - COG - Height
%161248.237,5520.17000858,00134.00310710,54.775,580051.32,6100123.86,8.44
%161249.001,5520.17000347,00134.00310369,54.777,580051.33,6100123.85,8.44
%Input file names: X18_Tide_20220715_0000.log; X18_Tide_20220716_0000.log
%The date must be in the file name's position: fName(end-16:end-9)
%==== Input format 2 (USV data format) >>>
%#<Time> HHmmss.zzz,<Date> dd,<Date> MM,<Date> yyyy, Latitude WGS-84 - COG - Latitude WGS-84, Longitude WGS-84 - COG - Longitude WGS-84, Height WGS-84 - COG - Height WGS-84, Easting - COG - Easting, Northing - COG - Northing, Height - COG - Height
%000001.002,19,07,2022,5516.94135841,00117.68878421,46.600,608305.37,6127537.33,0.87
%000002.002,19,07,2022,5516.94081535,00117.68899275,46.623,608305.17,6127536.32,0.89
%Input file names: X18_Tide_20220715_0000.log; X18_Tide_20220716_0000.log
%==== Input format 3 >>>
%23:27:38,2.39
%23:28:08,2.60
%Input file names: tide_innomar_20220512.txt; tide_innomar_20220513.txt
%The date must be in the file name's position: fNameN(end-11:end-4)
%==== Output format 1 (DugInsight data format) >>>
%1612:00 15 7 2022 4.74
%1613:00 15 7 2022 4.74
%==== Output format 2 (Silas tdx format) >>>
%08:40 1.974
%08:41 2.007
%The files create for each day separately. Output file names example: Tide_20220512.tdx; Tide_20220513.tdx
%==== Commands example >>>
%{'TidesRead','d:\x18_data\x18_raw_tides\01\','1'};Tides4Soft; -- Read tides files from folder;
%{'TidesDecimate',10};Tides4Soft; -- Decimate tides for quick smoothing (use each 10 values only);
%{'TidesDespSmoothInterp',600};Tides4Soft; -- Manual despiking, smoothing in a window 600 points and interpolation for gaps;
%{'TidesWrite','d:\x18_data\x18_raw_tides\out.txt'};Tides4Soft; -- Create and write file for DugInsight with data-per-minute
%==== Keys for manual despiking (use icon to select edit tool) >>>
%LeftMouseButton - first selection element (first part of rectangle or first point at curve);
%RightMouseButton - second selection element (second part of rectangle or second point at curve);
%MiddleMouseButton - "delete" points in selected area (set to NaN Y-coordinate value);
%z - undo; x - redo; q - Exit from edit mode. Backspace (or any key) - finish manual despiking.

gKey=ans;
if strcmp(gKey{1},'TidesRead') %Read tides files from folder
    %{'TidesRead','d:\x18_data\x18_raw_tides\01\','1'};Tides4Soft;
    %{'TidesRead','d:\tide_innomar_orig3\','3'};Tides4Soft;
    try fName=gKey{2};catch,fName=input('Input folder name=');end; try ff=gKey{3};catch,ff=input('Input format type=');end
    dz=dir(fName);dz([dz.isdir])=[];fName=[repmat(fName,length(dz),1) char(dz.name)];TD=[];
    for nn=1:size(fName,1)
        fNameN=deblank(fName(nn,:));
        switch ff
            case 1
```

```

[~,tmp]=gDataTxtRead(fNameN,1,7,'%f',' ','\r\n');
GpsTime=gNavTime2Time('HMS2Sd',tmp(:,1));GpsDay= repmat(gNavTime2Time('YMD2Dx',str2num(fNameN(end-16:end-9))),size(GpsTime));
GpsLat=gNavAng2Ang('DM2D',tmp(:,2));GpsLon=gNavAng2Ang('DM2D',tmp(:,3));

TD0=struct('GpsDay',GpsDay,'GpsTime',GpsTime,'GpsLat',GpsLat,'GpsLon',GpsLon,'GpsHgtGeoid',tmp(:,4),'GpsE',tmp(:,5),'GpsN',tmp(:,6),'Tide',tmp(:,7));
);
    case 2
        [~,tmp]=gDataTxtRead(fNameN,1,10,'%f',' ','\r\n');
        GpsTime=gNavTime2Time('HMS2Sd',tmp(:,1));GpsDay=gNavTime2Time('YMD32Dx',tmp(:,4),tmp(:,3),tmp(:,2));
        GpsLat=gNavAng2Ang('DM2D',tmp(:,5));GpsLon=gNavAng2Ang('DM2D',tmp(:,6));

TD0=struct('GpsDay',GpsDay,'GpsTime',GpsTime,'GpsLat',GpsLat,'GpsLon',GpsLon,'GpsHgtGeoid',tmp(:,7),'GpsE',tmp(:,8),'GpsN',tmp(:,9),'Tide',tmp(:,10));
);
    case 3
        fid=fopen(fNameN,'r');tmp=fscanf(fid,'%d:%d:%d,%f',[4 inf]);fclose(fid);
        GpsTime=gNavTime2Time('HMS32Sd',tmp(1,:),tmp(2,:),tmp(3,:));GpsDay= repmat(gNavTime2Time('YMD2Dx',str2num(fNameN(end-11:end-4))),size(GpsTime));
        TD0=struct('GpsDay',GpsDay,'GpsTime',GpsTime,'Tide',tmp(4,:));
        otherwise,error('Incorrect format number!');
    end
    if isempty(TD),TD=TD0;else,TD=gFieldsRowAppend(TD,TD0,numel(TD.GpsTime));end
end
[TD.GpsDay,TD.GpsTime]=gNavDayCheck(TD.GpsDay,TD.GpsTime);
[~,TD.t]=gNavTime2Time('DxSd2DmS',TD.GpsDay,TD.GpsTime);L=[false diff(TD.t)==0];TD=gFieldsRowSet(TD,numel(TD.GpsTime),L,[]);
TD.Mask=true(size(TD.GpsTime));TD.TideSm=nan(size(TD.GpsTime));TD.TideSmInt=nan(size(TD.GpsTime));
figure(100);plot(TD.t,TD.Tide,'.b');
clearvars fName fid ff dz nn fNameN tmp GpsTime GpsDay GpsLat GpsLon TD0 L;
end
if strcmp(gKey{1},'TidesDecimate') %Use each N values only (decimate tides for quick smoothing)
    %{'TidesDecimate',10};Tides4Soft;
    try dK=gKey{2};catch,dK=input('Decimate step=');end
    TD=gFieldsRowGet(TD,numel(TD.GpsTime),1:dK:numel(TD.GpsTime));
    figure(100);plot(TD.t,TD.Tide,'.b');
    clearvars dK;
end
if strcmp(gKey{1},'TidesDespSmoothInterp') %Despike manually, smoothing and interpolation
    %{'TidesDespSmoothInterp',600};Tides4Soft;
    try smW=gKey{2};catch,smW=input('Smooth window size=');end
    a=figure('Name','Despike','NumberTitle','off');p=gMapPickHandleNan(TD.t,TD.Tide,a);pause;mask=~get(p,'UserData');close(a);TD.Mask(mask)=false;
    TideSm=smooth(TD.t(TD.Mask),TD.Tide(TD.Mask),smW,'lowess');TD.TideSm(TD.Mask)=TideSm;
    TD.TideSmInt=interp1(TD.t(TD.Mask),TD.TideSm(TD.Mask),TD.t,'spline','extrap');
    figure(100);plot(TD.t,TD.Tide,'.b');hold on;plot(TD.t,TD.TideSm,'.r');plot(TD.t,TD.TideSmInt,'-c');hold off;
    clearvars smW a p mask TideSm;
end
if strcmp(gKey{1},'TidesWrite') %Create and write file for software with value-per-minute
    %{'TidesWrite','d:\x18_data\X18_raw_tides\out.txt',1};Tides4Soft;
    %{'TidesWrite','d:\tide_innomar_orig4\',2};Tides4Soft;
    try fName=gKey{2};catch,fName=input('Output file or folder name=');end; try ff=gKey{3};catch,ff=input('Output format type=');end
    t=fix(TD.t(1)./60)*60:60:TD.t(end);[~,di]=min(abs(t-TD.t(1)));d=TD.GpsDay(di);
    TideSmInt=interp1(TD.t,TD.TideSmInt,t,'spline','extrap');figure(100);plot(TD.t,TD.Tide,'.b');hold on;plot(t,TideSmInt,'.r');
    [GpsDay,GpsTime]=gNavTime2Time('Dms2DxDs',d,t);

```

```

switch ff
case 1
[Y,Mm,D]=gNavTime2Time('Dx2YMD3',GpsDay);[H,M,S]=gNavTime2Time('Sd2HMS3',GpsTime);
fid=fopen(fName,'w');fprintf(fid,'%02d%02d:%02d\t%d\t%d\t%d\t%0.2f\t\t\t\r\n',[H;M;S;D;Mm;Y;TideSmInt]);fclose(fid);
case 2
L=[1 find(diff(GpsDay))+1 numel(GpsDay)+1];
for nn=1:numel(L)-1
[YMD]=gNavTime2Time('Dx2YMD',GpsDay(L(nn)));[H,M,S]=gNavTime2Time('Sd2HMS3',GpsTime(L(nn):L(nn+1)-1));
fid=fopen([fName '\Tide_' num2str(YMD) '.tdx'],'w');fprintf(fid,'%02d:%02d %0.3f\r\n',[H;M;TideSmInt(L(nn):L(nn+1)-1)]);fclose(fid);
end
otherwise,error('Incorrect format number!');
end
clearvars fName ff t di d TideSmInt GpsDay GpsTime Y Mm D H M S L nn YMD fid;
end

```

%mail@ge0mlib.com 13/08/2022