

Sediment Load Estimation General Instructions

Document Format Information: **R commands are in bold text.** Important output from *R* commands that should be saved to the *doc* file (see next section) is shown in **Bold Courier Font**. Comments intended to go in the *doc* file are in **bold** text preceded by the # symbol. Explanations of the steps, i.e., comments in this file that are NOT intended to go into the *doc* file are in this plain text. Names of programs, files, objects, formats, and jargon are *italicized*.

1. R Template document: the “*doc*” file. Begin by opening the template file [stnhy.doc.rtf](#). Choose ‘save as’ and name the *doc* file after your station and hydro year. This document will serve as a working template of *R* commands, so the commands given in this example can be cut and pasted from the template rather than typed manually. The *doc* file is also a place to document the *R* commands, relevant output, and comments about the completed load estimate.
2. In the *doc* file, under the edit menu, choose *Replace*. Replace *stn* with your station code and select *Replace All*. Replace *hy* with your two-digit hydro year. Replace *Storm nnn* with your storm number, and insert the storm dates and times in the following format:
Storm *nnn*: *yymmdd, hhmm, yymmdd, hhmm*

The template (*doc* file) now contains most (but not all) of the commands and options that may be used to estimate sediment loads. You will use some of the commands in the template, and delete those you determine un-necessary from your final record.

3. Install necessary folders and files. Instructions for setting up the file structures needed to run *ttsAdjuster* are found in the [ttsAdjuster User’s Manual](#). *R* procedures require the same organization, but require only a *flo* file (electronic data) and an *isc* file (laboratory data).
4. Familiarize yourself with the storm event, if you are not already. *ttsAdjuster* can be used to view the stage and turbidity data, as well as the timing of the samples, and the suspended sediment concentration (SSC) of each sample.

For small creeks, it can be common for the turbidity probe not to be submerged until some hours after a storm begins, leading to questionable or bad turbidity data. Any storm event can contain periods of questionable or bad turbidity data. The exact start and end times of these periods will be essential information, because turbidity cannot be used as the surrogate for SSC if any corrected turbidity data are given as “NA” in the *flo* file. Periods of bad turbidity data have a turbidity code of 2 in the *flo* file and are displayed with a horizontal line red on the time axis in *ttsAdjuster*. Periods of questionable turbidity data have a turbidity code of 1 in the *flo* file and are displayed with a gray line in *ttsAdjuster*. While questionable turbidity can be used as the surrogate for SSC, this may not be the best choice so it is important to know the exact start and end times for questionable turbidity periods as well. The exact times of these periods can be obtained from the *ttsAdjuster* plot by using the *XY monitor* and the mouse. You may also read the field forms to see if they contain any relevant information such as problems with samples (spillage during transport or cross contamination). Check the lab data for similar notes regarding any issues with the samples or their processing.

Briefly summarize the situation in the *doc* file. Begin your comments with a pound sign # so they can be seen as comments rather than *R* commands or output. For example:

```
# dd number, sample numbers: any problems with samples during processing
# Turbidity and flow data issues with dates and times
```

5. Start an *R* session and read the corrected data for one water year from the *flo* file to an *R* object called *stnHY.flo* using the following command, or copy and paste it from the *doc* file. Notice the command below uses the symbols less than and minus <- where the template uses an underscore _ instead. These are functionally the same in versions of *R* before 1.8.0. Later versions do not recognize underscore for assignments.

```
stnHY.flo<-read.flo("stn",HY)
```

6. Read the lab data into an *R* object called *stnHY.lab* by typing the following command, also available in the template.

```
stnHY.lab<-read.lab("stn",HY)
```

7. To check for any inconsistencies between the lab data and the data in the *flo* file, use the following command:

```
mismatches("stn",HY)
```

8. Merge the electronic data with each sediment sample into an *R* object called *stnHY.sed* using the following command, available in the template.

```
stnHY.sed _ merge.flo("stn",HY)
```

NOTE: if you make any adjustments to the data (the *flo* file) while in *ttsAdjuster*, you must repeat steps 5 and 8 above so the corrected data will be available in *R*. If you make any adjustments to the lab data (the *isc* file), you must repeat steps 6 and 8.

9. You will now create plots of suspended sediment concentration. Start by plotting SSC versus turbidity whenever turbidity data are available. You will use the *R* function called *turbsscplot*, and the data will be plotted using four model types: *linear*, *logxy*, *sqrt*, and *power*. This is the basic starting point in sediment load estimation, and so these are the first four commands in the template. (Other model types that can be considered as well are *loess* and *logx*.) Since you wish to compare these four plots with each other, you begin with the *R* command: **four()**. This command tells *R* to display the four plots together on one page.

Note that there are two ways to designate in the *R* commands which samples you wish to include in the plots. One is to specify the date range containing those samples (see the first two commands below). The second is to specify the samples you wish to include by their dump and bottle number (third and fourth commands below). Take note of which plot shows the strongest relationship before continuing to the next step. When new plots are created the old plots are gone, unless you have printed them out or copied them to a word document.

```
turbsscplot("stn",HY, yymmdd,time,yymmdd,time)
```

```
turbsscplot("stn",HY, yymmdd,time,yymmdd,time,type="logxy")
```

```
turbsscplot("stn",HY,dump=#,bot=#:#,type="sqrt")
```

```
turbsscplot("stn",HY,dump=#,bot=#:#,type="power")
```

If the turbidity data are not available, or if the above plots show a poor relationship between turbidity and SSC, you will next try discharge as the surrogate variable. In this case you use the *R* function *qsscplot*. Note that the default model type is "linear" for *turbsscplot*, but "logxy" for *qsscplot*.

```
qsseplot("stn",HY, yymmdd,time,yymmdd,time)  
qsseplot("stn",HY, yymmdd,time,yymmdd,time,type="linear")  
qsseplot("stn",HY, yymmdd,time,yymmdd,time,type="sqrt")  
qsseplot("stn",HY, yymmdd,time,yymmdd,time,type="power")
```

10. Use the plots to decide which model to try, leaving as much explanation as you like in your *doc* file. More than one model might be needed to accurately estimate the sediment load for a storm. Turbidity is almost always a better surrogate for SSC than discharge. Time can be a better surrogate than discharge for periods of bad, poor or missing turbidity data if the samples were collected frequently enough to define the shape of the turbidigraph.

Look back at a plot of the storm's stage and turbidity data using *ttsAdjuster* to see if it makes sense that the relationship has shifted at a certain point. If it does, you may use two separate models. It is not uncommon for the two limbs of the hydrograph to have different relationships between SSC and either turbidity or discharge. This will make more sense when working with a real example.

The periods you divide the storm into may be indicated by adjusting the time and/or bottles specified in the *turbsscplot* or *qsscplot* functions.

Remember to put a copy of the *R* commands you use into the *doc* file. A useful strategy is to put every command used in the *doc* file to start with, and then when the estimate is complete, delete the commands and comments that were not used to decide upon the final method.

11. To actually estimate the sediment load for this storm, custom *R* functions are used to construct data objects. You may think of a data object as a set of organized data like a small spreadsheet. We will construct data objects which contain the date/time of each 10 minute record, the surrogate variable value for each record, and the estimated or predicted SSC for each record.

The estimated/predicted SSC is calculated using the surrogate variable relationship selected, and indicated in arguments to the *R* routine. *turbsrc* is the function that makes a data object with turbidity as the surrogate, *flowsrc* makes a data object with discharge as the surrogate, and *lineartime* makes a data object with time as the surrogate.

The whole storm period needs to be covered by one or more such data objects. Each data object can use only one type of model/relationship based on one set of samples. If the storm is to be divided and estimated using more than one method or sample set, separate data objects must be created to represent each sub-period of the whole storm event.

These data objects should be continuous with regard to the data interval, and as continuous as possible with regard to the predicted SSC from the end of one to the start of the next. This means that a data object should start at the next interval past the end of the data object representing the previous sub-period, and the first predicted SSC of a data object should be close to the last predicted SSC of the data object representing the previous sub-period. In periods of rapidly changing SSC, the appropriate differences in SSC between consecutive intervals may be non-zero, reflecting the current rate of change of the SSC. Adjusting the exact time of the transition from one data object to the next will result in different beginning and ending predicted SSC values. This will all be described next.

12. When using two or more models to estimate the sediment load of a storm, choosing the correct time to transition between the models is important. To determine the best point in time to separate two periods of a storm, we begin by using *R* routines to create two overlapping *R* data objects.

A data object created using the routine *turbsrc* estimates sediment load using turbidity as the surrogate variable. This object can be named anything you find helpful. It is best to use a simple naming convention such as *turb*, *flow* or *time* followed by a number. You might name a turbidity based object for storm number 22 “turb22”. For this example, the first turbidity-based object is named *turb1*. The *R* command format is as follows: the name you choose is first, followed by the assignment operator, and the *R* function. The station, year, and start and end points are specified as comma-separated function arguments in parentheses.

```
turb1<-turbsrc(“stn”,HY, yymmdd,time,yymmdd,time)
```

This command will create the object *turb1* using the data from all the sample bottles which were taken during the time period specified in the command. You may wish to exclude the data from a bottle, for example if there was some problem with the sample processing which led to inconsistent results. You may also wish to include the data from a bottle outside that specified time period. In either case, the bottles you wish to include may be specified by including the dump number and bottle number(s) in the command, in addition to the dates and times specifying the estimation period. If more than one dump number is to be included, the *dump* and *bot* arguments should be vectors of equal length. The *c* and *rep* functions in *R* are useful for specifying these vectors (see *R* documentation).

```
turb1<-turbsrc(“stn”,HY, yymmdd,time,yymmdd,time,dump=n,bot=n:n)
```

To create an object using one of the model types, the *type* argument is specified just as in the plots created earlier, for example:

```
turb1<-turbsrc(“stn”,HY, yymmdd,time,yymmdd,time,dump=n,bot=n:n,type=“logxy”)
```

We could use the routine *flowsrc* to create an object with the discharge as the surrogate variable in estimating sediment load. You might name the first flow-based object *flow1*.

```
flow1<-flowsrc(“stn”,HY, yymmdd,time,yymmdd,time)
```

In cases where the relationship of both turbidity and flow to SSC is very poor, the next option is to use time as the surrogate variable. This is only an option when there is an adequate coverage

of data from sample bottles collected in this period. The routine *lineartime* would be used to create an object you might wish to call *time1*.

```
time1<-lineartime("stn",HY, yymmdd,time,yymmdd,time)
```

To overlap two data objects, the first object's end time is extended past the start time of the second object. When one object is created with turbidity as the surrogate variable, in most cases that object should cover all the time when turbidity data are good. Recall that turbidity is usually the preferred surrogate variable.

13. Once two overlapping objects are created, a plot is created using the *ttsplot* function to see where they intersect. If they do intersect, this point will be the best place to transition from one sub-period to the next because the data objects' predicted SSC will be the closest in value at their intersection point.

```
ttsplot("car",04,turb1,flow1)
```

To find the exact transition point from one period to the next, it may be most efficient to print a hardcopy of the *tts* plot and use a triangle to locate the time of intersection.

Once the best available transition point between each period of the storm has been determined, new objects must be created with the correct start and end times. The objects should no longer overlap in time. Each object will end ten minutes before the next one begins. You can use the same object names; the most recently created version of an object will replace an older object of the same name.

ttsplot can be used to view all the data objects together.

```
ttsplot("stn",HY,object1,object2,etc)
```

To produce the estimated total sediment load and statistics for the entire storm the *total* command is used:

```
total(object1,object2,etc)
```

It is important to not skip this step because this command will check for time gaps and overlaps.

14. Finally, we can make an *R* object that represents the whole storm event as 10 minute predicted SSC values. A good convention for naming this object is *STNstorm#.ssc*. The *R* output *yhat* refers to the estimated total load.

```
STNstorm#.ssc<-tts.ssc(object1,object2,etc)
```

```
yhat: ##
```

It is useful to copy the output lines from both the *R* functions used above into the *doc* file. This is a confirmation that the correct *R* objects were used in the *tts.ssc* routine. The *yhat* value should match the total load value in the output from the *ttsplot* and *total* functions we ran just before. If they don't match then you'll know that you used the wrong objects.

15. Finally, check the *doc* file you have constructed to be sure it includes all the *R* commands, output, and your comments that are relevant to the final sediment load estimate. You should have deleted any template commands, etc. which you did not use.

For a detailed description of the *R* functions used in sediment load estimation, including many options that are not described in this tutorial, see the reference guide [R procedures.pdf](#).