A Guide to Processing UBeTube Data in R

Jeremy W. Schallner, Justin C. Johnson, C. Jason Williams, Amy C. Ganguli,

2020

Contents

1 Introduction

The Upwelling Bernoulli Tube (UBeTube) is a device used to measure plot-scale runoff (Figure 1; Stewart et al. 2015). Runoff is funnelled into a vertical tube with a slot machined into its side. The height of water flowing out of the slot is measured with a vented pressure transducer placed within a stilling well at the bottom of the UBeTube. The flow rate can then be calculated as a function of the height measurement. Stewart et al. (2015) originally designed and tested the UBeTube and are an excellent reference for further information. An assessment of the UBeTube under conditions typical of rangelands can be found in Schallner et al. (2021).

The goal of this document is to provide an example of how to calibrate the UBeTube and then process a time series of UBeTube measurements. The resulting processed data will have converted water height measurements (mm) to flow rate into the UBeTube ($L \cdot min^{-1}$). An EM50 data logger (METER Group, Inc.) and a HYDROS 21 pressure transducer (METER Group, Inc.) are used in this example. An R script (*UBeTube_Processing.r*) is provided as a template that can be modified to account for different slot geometries and/or different data logger output file formats. To ensure the working directory is properly set, it is advisable to first open the included R project file, *UBeTubeR_REM.Rproj*, and then open the R script, *UBeTube_Processing.r.*

2 Calibration methods

Stewart et al. (2015) suggested a physical relationship between water level height and flow rate determined by Bernoulli's equation and parameterized using the dimensions of the slot machined into the side of the UBeTube. Although Stewart et al. (2015) provided a correction factor to account for observed bias of their physically-based estimates of discharge, Schallner et al. (2021) found this correction factor needed to be calibrated to minimize error. They also found estimates of discharge using a power-law relationship between water level height and flow rate were equally or more accurate than calibrated physically-based estimates of discharge using Bernoulli's equation. For simplicity, here we detail methods to calibrate using a power-law function and do not describe methods using Bernoulli's equation. Note Schallner et al. (2021) calibrated

Figure 1: UBeTube during calibration trials.

over flow rates ranging from 2.5-41.5 L $\cdot$ min^{-1} and using the slot geometry shown in Figure 2. Calibrations of UBeTubes with differing slot geometries or flow rates exceeding 41.5 L $\cdot$ min^{-1} may require refinement of calibration methods detailed here. It may be advisable to modify the slot geometry to avoid sediment becoming lodged in the bottom of the slot or to optimize for expected flow rates.

3 Data processing

All data processing steps are shown using the open-source statistical software, R. It is beyond the scope of this tutorial to teach basic R programming. We recommend first becoming familiar with R and downloading an integrated development environment such as RStudio before attempting to use the R script (*UBeTube_Processing.r*). Although the script is written to be accessible to R programming novices, a basic understanding of R programming is still needed. There are many free online resources to help learn R programming basics.

3.1 Calibration

Before the UBeTube can be deployed, calibration is needed to establish a known relationship between the height of water within the UBeTube and the outflow of water through the slot machined into its side (Figure

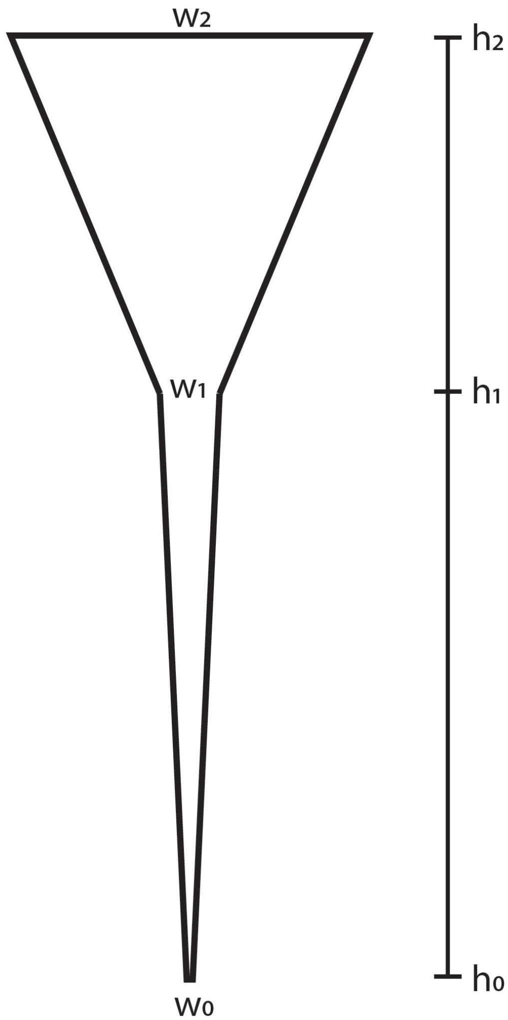

Figure 2: Slot geometry used in Schallner et al. (2021). Slot geometry may be optimized for expected flow rates or to limit sediment becoming lodged, although calibration methods may need to be revaluated depending on changes.

2). This can be achieved by taking paired measurements of both height and discharge across the expected range of flows to be measured in the field. We provide an example calibration dataset (*calibration_data.xls*) and the associated R script (*UBeTube_Processing.r*) in the supplemental materials to detail how to establish this relationship.

The calibration dataset needs to include three pieces of information for each paired measurement:

- *h0.mm* - The height (mm) of the base of the slot (h_0) relative to the pressure transducer. This can be determined by taking a height measurement with the pressure transducer when flow has ceased and the water level has stabilized at the bottom the slot (h_0).
- *hraw.mm* - The height (mm) of the water above the pressure transducer.
- *Q.Lmin* - The discharge ($\mathrm{L} \cdot \mathrm{min}^{-1}$) of water from the UBeTube measured independent of the pressure transducer. Possible measurement methods include timed samples of discharge or an inline flow meter. Flow should be at a steady state over the sampling period.

3.1.1 Import and wrangle data

To begin processing the calibration dataset (*calibration_data.xls*), the data must be read into R. We will use the *readxl* and *tidyverse* packages to load and wrangle the data. We will also use the *broom* package to more easily reference the model outputs used during calibration. If you haven't previously, you will need to install the packages via the *install.packages* function. Once installed, load each package using the *library* function.

```
#install.packages(readxl)
#install.packages(tidyverse)
#install.packages(broom)
library(readxl)
library(tidyverse)
library(broom)
```

We then read *calibration_data.xls* into R via the *read_excel* function. If your data file is a *.csv*, the function *read_csv* would allow you to import your data into R. We skip over the column headers and process the first tab. The calibration dataset has three columns named: *h0.mm*, *hraw.mm*, and *Q.Lmin*. The resulting tibble is named *calibration*. With the *mutate* function, we create the column, *h.mm*, which is the height of the water relative to the bottom of the slot. This is calculated by subtracting *h0.mm* from *hraw.mm*. We then convert *h* from mm to cm and store the output in the column named *h.cm* (Table 1).

```
#Read in calibration data
calibration <- read_excel("calibration_data.xls",
                        sheet=1,
                        skip=1,
                        col_types = c("numeric", "numeric", "numeric"),
                        col_names=c("h0.mm", "hraw.mm", "Q.Lmin"))


#Wrangles data
calibration1 <- calibration %>%
  mutate(h.mm = hraw.mm-h0.mm, #Calculates height of water relative to bottom of slot
         h.cm = h.mm/10)       #Converts h to cm
```

Table 1: Example of a portion of calibration data after data wrangling.

h0.mm	hraw.mm	Q.Lmin	h.mm	h.cm
360	410	2.574956	50	5
360	410	2.571177	50	5
360	410	2.551640	50	5

3.1.2 Power-law rating curve

Using the calibration dataset, we fit the following power-law function to the height and discharge data:

$$Q = ah^b \tag{1}$$

where Q is the discharge from UBeTube ($\mathrm{L \cdot min^{-1}}$), h is the height of water above the bottom of the slot (cm), and a and b are fitted scale and shape parameters, respectively.

To fit the power-law function to the calibration data, we use the *nls* function, which provides non-linear least-squares estimates of a and b. The first argument is the formula of the model, followed by the dataset to be used, and finally the starting values to begin estimating a and b. The resulting model object (*ratingcurve.model*) is tidyed into a tibble (*ratingcurve.tidymodel*) using the *tidy* function. This allows the fitted parameters of the model to be more easily referenced in future calculations. To visually assess the fit of the power-law function, we can plot our model with the calibration data using the *ggplot* function, which is part of the *ggplot2* package already loaded within the *tidyverse* meta-package. *ggplot* has relatively intuitive syntax and creates readily customizable and elegant figures. For brevity, we suggest referencing RStudio's *ggplot* cheatsheet for further information on the syntax used here.

```
#Runs power regression to predict discharge from stage.
ratingcurve.model <-nls(calibration1$Q.Lmin~a*calibration1$h.cm^b,
                        data=calibration1,
                        start=list(a=1,b=1))
ratingcurve.tidymodel <- tidy(ratingcurve.model) #Tidys model

#Extracts a and b estimates
a <- as.numeric(ratingcurve.tidymodel[1,2])
b <- as.numeric(ratingcurve.tidymodel[2,2])

#Creates label for plot with model results
label <- sprintf("Q == %.3g*h^%.3g", a, b)

#Plots rating curve
ggplot(calibration1, aes(h.cm, Q.Lmin))+
  geom_point()+
  labs(x="h (cm)",
       y=bquote('Q (L·'*min^-1*')'))+
  geom_smooth(method = 'nls',
              formula = 'y~a*x^b',
              method.args = list(start= c(a = 1,b=1)),
              se=FALSE)+
  theme_classic()+
  annotate("text",
           x=0.6*median(calibration1$h.cm, na.rm=TRUE),
           y=median(calibration1$Q.Lmin, na.rm=TRUE),
           label= label,
           color="blue",
           parse=TRUE)
```

3.2 Time series

When deploying the UBeTube in the field, a time series of height measurements is collected, which can then be processed to determine the runoff rate into the UBeTube. Again, it is important to determine the height of the bottom of the slot relative to the pressure transducer (h_0). This can be achieved using the same methods as discussed in the calibrations. It may be advisable to determine h_0 each time you maintain the UBeTube in the field to limit potential error caused by pressure transducer drift. This is analagous to taring a scale after repeated use. In this example, an EM50 data logger (METER Group, Inc.) and a HYDROS 21 pressure transducer (METER Group, Inc.) are again used. Procedures for importing data may differ when using alternative equipment.

3.2.1 Import and wrangle data

An example time series dataset is provided in the supplemental materials (*example_dataset.xls*). The EM50 data logger outputs a *.xls* file with two tabs and three rows of headers. Again, we use the *read_excel* function to read in the data. We skip over the headers and process the first tab. The HYDROS 21 pressure transducer also measures temperature and electrical conductivity, which we will exclude from the dataset. The resulting tibble is named *UBeTube*. It has two columns, the time and date of each measurement (*time*) and the associated height (mm) of the water above the pressure transducer (*hraw.mm*). With the *mutate* function, we add a column that converts height from mm to cm (*hraw.cm*), and we add the column *time.sec*, which is the *time* column converted to seconds (Table 1).

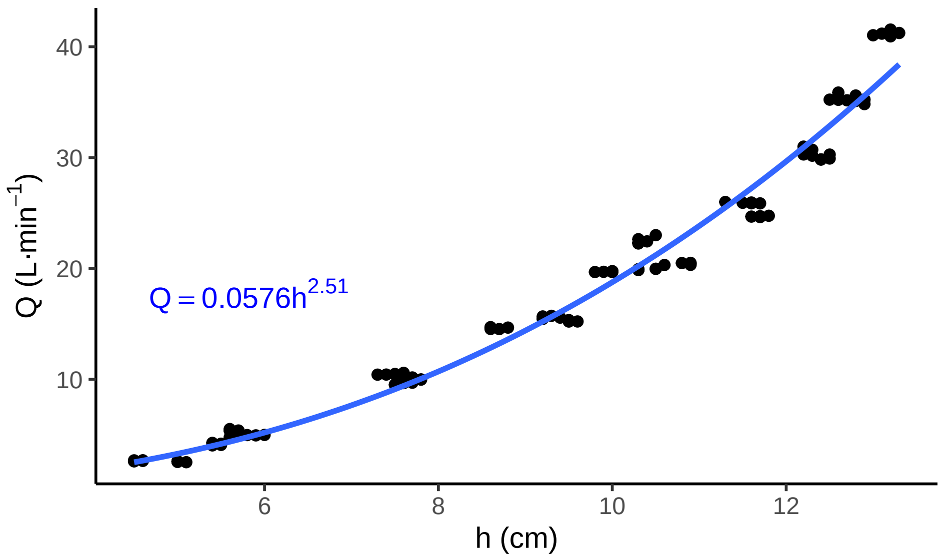

Figure 3: An example of a rating curve used to predict discharge from the height of water within the UBeTube.

```
h0 <- 35.7 #Height of h0 relative to pressure transducer

#Reads in time series dataset
UBeTube <- read_excel("example_dataset.xls",
                        sheet=1,
                        skip=3,
                        col_types = c("date", "numeric", "skip", "skip"),
                        col_names=c("time", "hraw.mm"))

UBeTube1 <- UBeTube %>%
  mutate(hraw.cm = hraw.mm/10, #Converts height to cm from mm
         time.sec = as.numeric(time, unit='sec'), #Converts time to seconds
         h.cm = hraw.cm-h0) #Determines height of water in UBeTube relative to h0
```

Table 2: Example data set after data wrangling.

time	hraw.mm	hraw.cm	time.sec	h.cm
2019-08-08 10:14:00	361	36.1	1565259240	0.4
2019-08-08 10:15:00	362	36.2	1565259300	0.5
2019-08-08 10:16:00	362	36.2	1565259360	0.5

3.2.2 Outflow calculations

To estimate the flow of water leaving the UBeTube (Q) at a given time, we use Equation (1) to relate h (cm) to Q ($\mathrm{L} \cdot \mathrm{min}^{-1}$).

```r
UBeTube2 <- UBeTube1 %>%
  mutate(Q.Lmin=a*h.cm^b) #Calculates outflow based on Equation 1
```

3.2.3 Inflow calculations

The flow rate equations do not account for changes in storage within the UBeTube. Storage can be calculated by:

$$S_t = \frac{\pi d^2 h_t}{4} \tag{2}$$

where S_t is storage at time t, d is the interior diameter of the UBeTube, and h_t is the height of the water at time t. To calculate inflow into the UBeTube, the rate of change in storage can then be added to the average flow rate over a period time, calculated as:

$$I_t = (\frac{S_t - S_{t-1}}{\Delta t}) + (\frac{Q_t + Q_{t-1}}{2}) \tag{3}$$

where I_t is inflow into the UBeTube at time t, Δt is the difference in time between t and the previous measurement *t-1*, and Q_t is the flow rate at time t.

```r
d <- 10.16 #Interior diameter of UBeTube (cm)

#Calculates inflow (L/min)
UBeTubeFinal <- UBeTube2 %>%
mutate(S.cm3s = (pi*(d^2)*h.cm)/4, #Calculates volume of water stored in UBeTube
       S.Lmin = S.cm3s*60/1000, #Converts storage to L/min
       I.Lmin = (S.Lmin-lag(S.Lmin))/(time.sec-lag(time.sec))
         +((Q.Lmin+lag(Q.Lmin))/2)) #Calculates inflow using equation 3
```

Table 3: Example data set after converting height (h.cm) to inflow (I.Lmin).

time	hraw.mm	hraw.cm	time.sec	h.cm	Q.Lmin	S.cm3s	S.Lmin	I.Lmin
2019-08-08 12:51:00	473	47.3	1565268660	11.6	27.23663	940.4491	56.42694	27.23663
2019-08-08 12:52:00	473	47.3	1565268720	11.6	27.23663	940.4491	56.42694	27.23663
2019-08-08 12:53:00	472	47.2	1565268780	11.5	26.65048	932.3418	55.94051	26.93544

3.3 Hydrograph

After calculations have been completed, it is often useful to plot a hydrograph to visualize the data. This can be achieved using the *plot* or *ggplot* functions.

```r
plot(UBeTubeFinal$time,UBeTubeFinal$I.Lmin, type="l",
     xlab="Time (hh:mm)",
     ylab= expression('Inflow (L·min' ^ {-1}*')'))
```

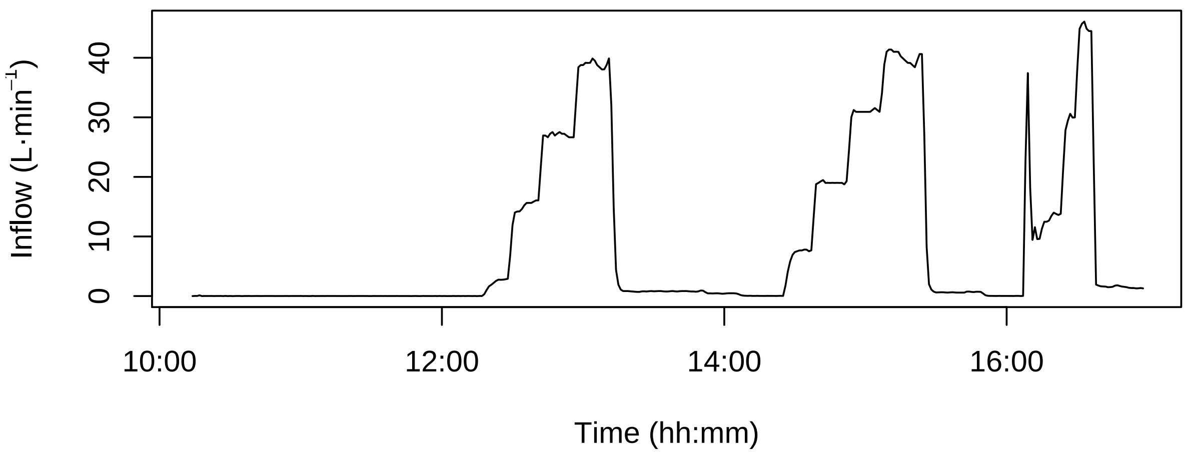

Figure 4: An example of an UBeTube hydrograph after data has been processed. Example data are from calibration trials.

References

Schallner, J. W., J. C. Johnson, C. J. Williams, and A. C. Ganguli. 2021. "Evaluation of a runoff monitoring methodology for rangelands: UBeTubes." *Rangeland Ecology and Management*.

Stewart, R. D., Z. Liu, D. E. Rupp, C. W. Higgins, and J. S. Selker. 2015. "A new instrument to measure plot-scale runoff." *Geoscientific Instrumentation, Methods and Data Systems* 4 (1): 57–64. https://doi.org/10.5194/gi-4-57-2015.