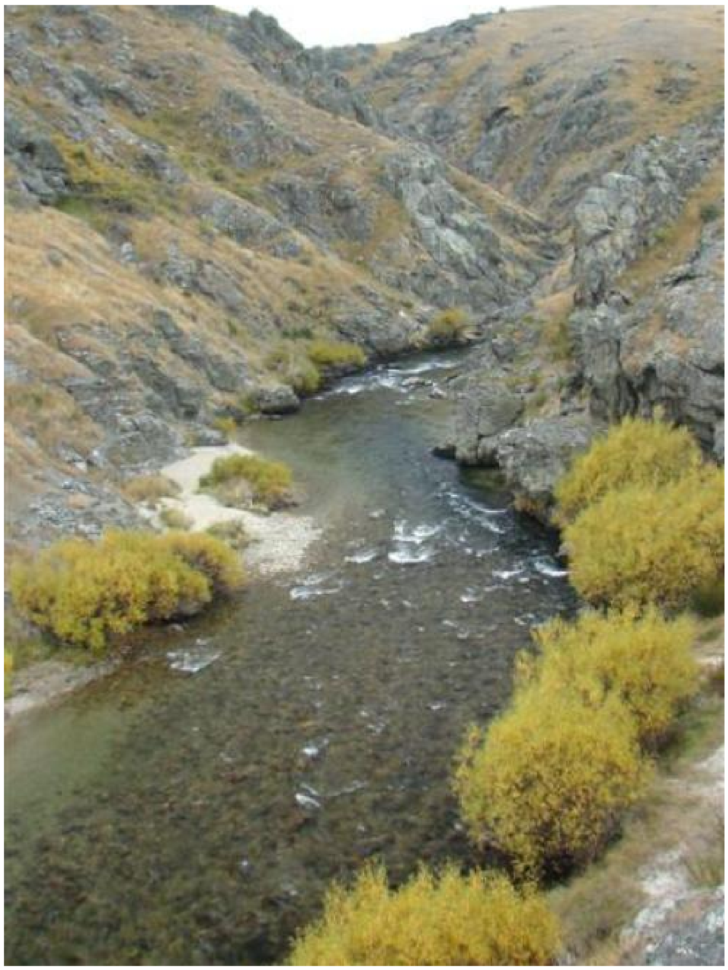
CE 5362 Surface Water Modeling
Lesson 7

by
Theodore G. Cleveland
Department of Civil, Environmental, and Construction Engineering
Texas Tech University



Contents

1 Steady Water Surface Profiles - II

The last lesson concluded with an example of computing a backwater curve using a variable spatial step. This lesson examines the fixed spatial step.

1.1 Finite Difference Methods — Fixed Depth Change Step Method

The fixed-step refers to fixed changes in depth for which we solve to find the variable spatial steps. The method is a very simple method for computing water surface profiles in prismatic channels. A prismatic channel is a channel of uniform cross sectional geometry[1] with constant bed (topographic) slope.

For the variable step method, the momentum equation is rewritten as a difference equation (after application of calculus to gather terms) then rearranged to solve for the spatial step dimension [2].

$$\frac{\frac{V_{i+1}^2}{2g} - \frac{V_i^2}{2g}}{\Delta x} + \frac{h_{i+1} - h_i}{\Delta x} = S_o - \bar{S_f} \tag{1}$$

where $\bar{S_f}$ is the average slope of the energy grade line between two sections (along a reach of length Δx, the unknown value).

Rearrangement to isolate Δx produces an explicit update equation that can be evaluated to fond the different values of Δx associated with different flow depths. The plot of the accumulated spatial changes versus the sum of the flow depth and bottom elevation is the water surface profile.

$$\frac{(h_{i+1} + \frac{V_{i+1}^2}{2g}) - (h_i + \frac{V_i^2}{2g})}{S_o - \bar{S_f}} = \Delta x \tag{2}$$

The distance between two sections with known discharges is computed using the equation, all the terms on the left hand side are known values. The mean energy gradient ($\bar{S_f}$) is computed from the mean of the velocity, depth, area, and hydraulic radius for the two sections.

The friction slope can be computed using Manning's, Chezy, or the Darcy-Weisbach friction equations adapted for non-circular, free-surface conduits.

[1] Channel geometry is same at any section, thus rectangular, trapezoidal, and circular channels if the characteristic width dimension is constant would be prisimatic.

[2] The equation here is written moving upstream, direction matters for indexing. Thus position $i+1$ is assumed upstream of position i in this essay. This directional convention is not generally true in numerical methods and analysts need to use care when developing their own tools or using other tools. A clever analyst need not rewrite code, but simple interchange of upstream and downstream depths will handle both backwater and front-water curves.

Using the scripts from last lesson the following example was presented. An additional example is presented, then onto fixed spatial steps.

1.1.1 Example 1 — Backwater curve

Figure 1 is a backwater curve[3] for a rectangular channelwith discharge over a weir (on the right hand side — not depicted). The channel width is 5 meters, bottom slope 0.001, Manning's $n = 0.02$ and discharge $Q = 55.4\frac{m^3}{sec}$.

Using the `backwater` function and some plot calls in **R** we can duplicate the figure (assuming the figure is essentially correct).

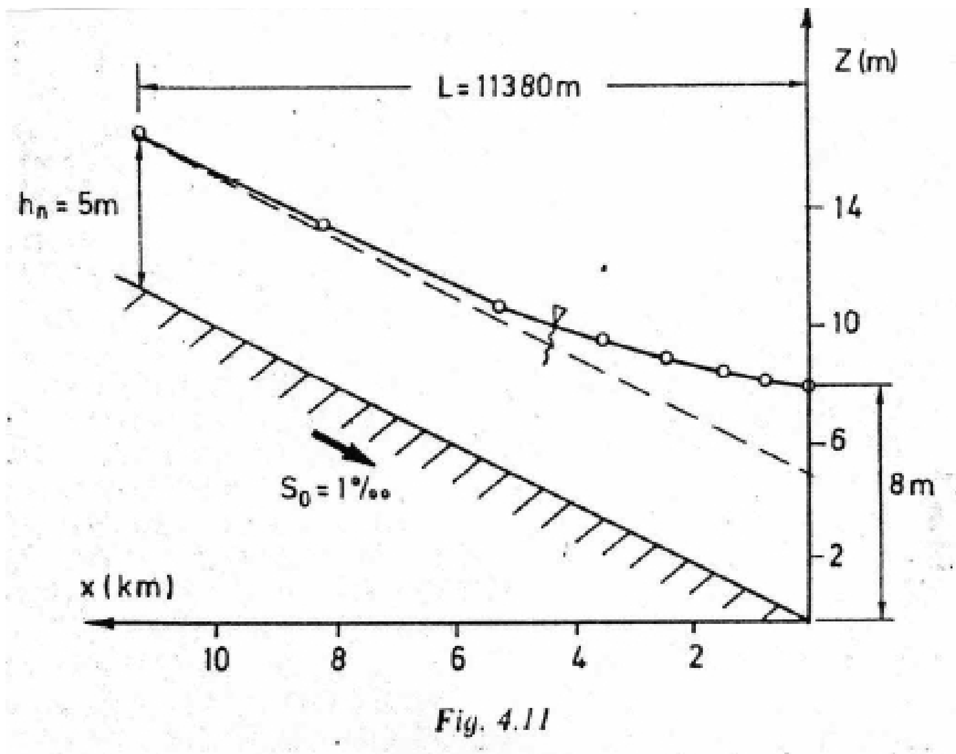


Figure 1. Example backwater curve.

[3]Page 85. Koutitas, C.G. (1983). Elements of Computational Hydraulics. Pentech Press, London 138p. ISBN 0-7273-0503-4

Listing 1. R Console Output when script is run

.

```
> # here is the function call
> backwater(begin_depth=8,end_depth=5,how_many=31,
    + discharge=55.4,width=5,mannings_n=0.02,slope=0.001)
          distance    depth        bse       wse
 [1,]       0.0000 8.000000  0.0000000  8.000000
 [2,]    -283.3504 7.806452  0.2833504  8.089802
 [3,]    -428.0112 7.709677  0.4280112  8.137689
 [4,]    -574.8516 7.612903  0.5748516  8.187755
 [5,]    -724.0433 7.516129  0.7240433  8.240172
 [6,]    -875.7785 7.419355  0.8757785  8.295133
... Many Rows ...
[29,]  -7186.0943 5.193548  7.1860943 12.379643
[30,]  -8389.5396 5.096774  8.3895396 13.486314
[31,] -11393.2010 5.000000 11.3932010 16.393201
```

Figure 2 is the same situation computed and plotted using the script in this essay[4].

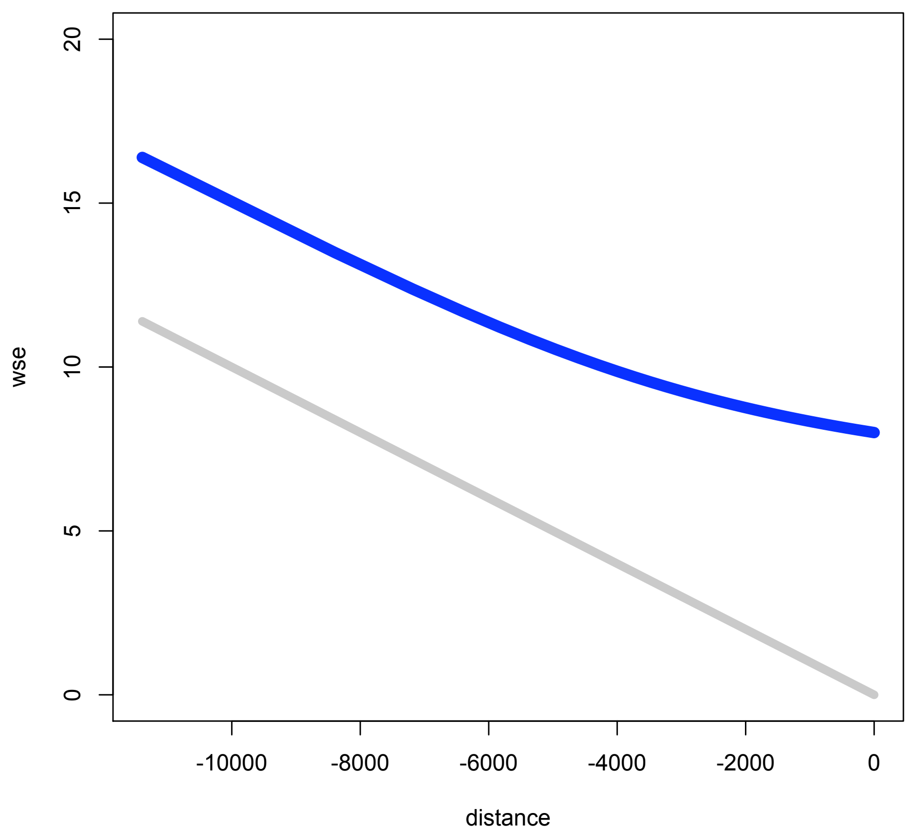


Figure 2. Backwater curve computed and plotted using **R**.

[4]With some additions that will be demonstrated in class!

1.1.2 Example 2 — Front-water curve

Figure 3 is another illustrative case. Here the water discharges into a horizontal channel at a rate of $1\frac{m^3}{sec}$ per meter width. Assuming Manning's $n \approx 0.01$ we wish to compute the profile downstream of the gate and determine if it will extend to the sharp edge[5].

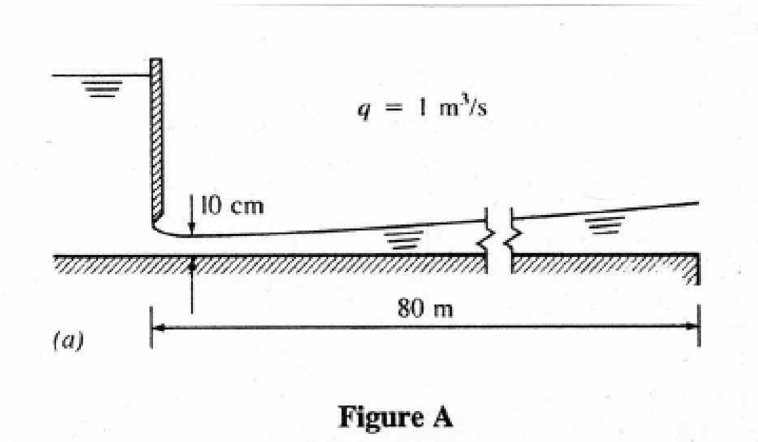


Figure 3. Flow under a sluice gate.

We would need to know the critical depth for the section ($\approx 0.47 meters$), then compute the profile moving from the gate downstream (a frontwater curve with respect to the gate).

With the `backwater` function, all we really need to do is change the function arguments because it is already built for rectangular channels.

Observe that the distance is now incrementing forward (by choice of begin and end depths). Figure 4 is the situation computed and plotted using the script.

Listing 2. R Console Output when script is run

.

```
> # The function --- note the changes in parameter values!
> backwater(begin_depth=0.1,end_depth=0.47,how_many=20,
   + discharge=1,width=1,mannings_n=0.01,slope=0.000)
       distance  depth bse     wse
 [1,]  0.00000 0.1000   0 0.1000
... Rows ...
[13,] 79.00486 0.3405   0 0.3405
[14,] 82.82243 0.3590   0 0.3590
... Rows ...
```

[5]Obviously the profile will change a lot near the edge, but the question is will the profile continue to rise as depicted if the edge were further away?

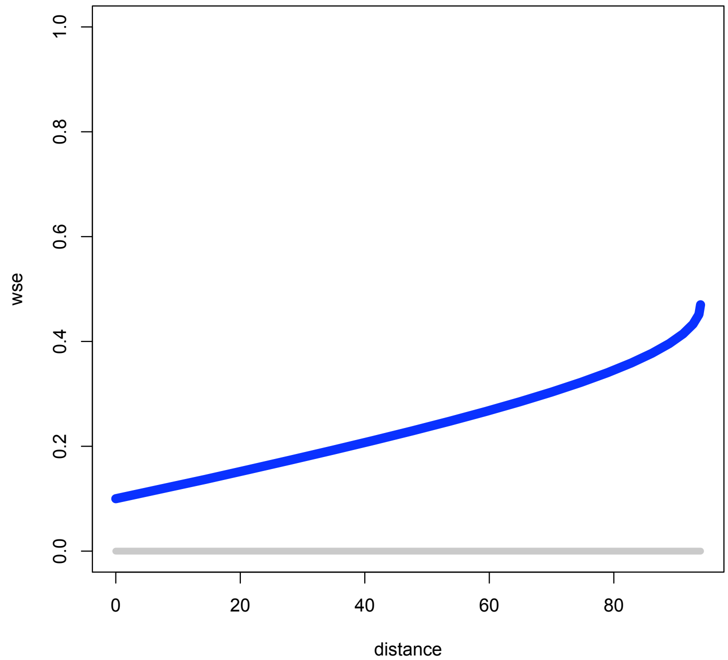


Figure 4. Frontwater curve computed and plotted using **R**.

1.1.3 Finite Difference Methods — Fixed Space Change Method

The fixed-depth change, variable-space result is a useful tool, but not terribly practical because we mostly perform engineering hydraulics calculations to estimate values (depth, pressure, force) at prescribed locations in space, so we need another approach to the problem where we can prescribe the spatial locations, and solve for the depths.

First the gradually varied flow equation is rearranged for relating the change in specific energy between two section to the spatial difference and the slope differences as

$$\Delta h_s = \Delta(h + \frac{V^2}{2g}) = \Delta x(S_o - S_f) \tag{3}$$

The computation of h_{i+1}, V_{i+1} from h_i, V_i is performed by iteration. An initial value for h_{i+1} that is known to be too large is used in Equation 3 along with the known value of h_i to compute a trial value $h_{s(i+1)}$.

Then the trial value is used in the right hand side of Equation 4

$$h_s = (h + \frac{V^2}{2g}) \tag{4}$$

The two trial values are compared and the next value of h_{i+1} is computed by sucessively decreasing until the two values computed by the difference equation and the definition of specific energy coincide. The example below uses a method from Hamming to make the comparisons and adjust the guesses until they are sufficiently close enough.

Example (Non-Prismatic Channel), Fixed Spatial Steps)
A plan view of a rectangular channel of variable width as shown in Figure 5.

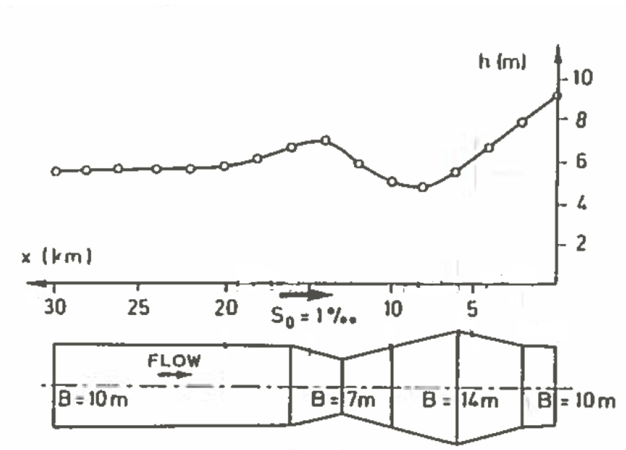


Figure 5. Non-Prismatic Rectangular Channel.

The channel conveys $Q = 100\ m^3/sec$, with a bottom slope of 0.001 and average Manning's n value of 0.033. A backwater curve is caused by a weir at the downstream end (to the right in the figure) by a 7 meter tall weir. Flow depth over the weir is at critical depth $h_c = 2.17$ meters. Normal flow in the upstream portion for 10-meter channel width is $h_n = 5.6$ meters. Using the fixed space step method determine and plot a profile view of the water surface and channel bottom.

Listing 3 is an entire listing for the example problem. The listing introduces our first intricate use of `for(...)`, `while(...)`, and the use of `break` to control the program logic. The outer `for(...)` loop is used to step through the various cross sections while the two inner `while(...)` "loops" are used to perform the specific energy balancing logic and exit under different conditions.

In older languages where `goto ... <label>` was allowed logical control crossed over loopback paths. Computer scientists claimed that a `goto ... <label>` was unnecessary – this example here was ported from a `FORTRAN` code from the 1980's, and indeed the `goto ... <label>` statements were not required. Figure 6 is a screen capture of the result for the example problem.

Listing 3. R script for Backwater Curve by Fixed Space Method

.

```
###### Fixed Space Step Method for Backwater Curve  ###########
#Uses Hamming's Method for Matching Specific Energy at Sections
################
imax <- 30              #number of computational stations
dx <- -1000             #spacing between stations
manningsN <- 0.033      #mannings n value
slopeChan <- 0.001      #channel slope
normalD  <- 5.6         #upstream station normal depth
controlD <- 9.17        #downstream station control depth
discharge <- 100        #steady discharge
#################
topwidth <- numeric(0) # allocate a topwidth vector
# populate the vector -- should use a file read for general program
topwidth <- c
    (10,10,10,11,12,13,14,13,12,11,10,9,8,7,8,9,10,10,10,10,10,10,10,10,10,10,10,10,10,10)
# populate the vector (should convert to a file read)
velo <- rep(0,imax) # allocate a velocity vector, put zeroes everywhere
velo[1] <- discharge/topwidth[1]/controlD #set velocity at control section
depth <- rep(0,imax) # allocate a depth vector, put zeroes everywhere
depth[1] <- controlD
spDepth <- rep(0,imax) # allocate a sp. depth vector, put zeroes everywhere
spDepth[1] <- ((velo[1]^2)/(2.0*9.81)) + depth[1]
spDepthT <- rep(0,imax) # allocate a sp. depth vector, put zeroes everywhere
#################
for(i in 2:imax){
  depth[i] <- 10
  difn <- 1.0
  dh <- 1.0
  while(dh > 0.0001){
    dif <- difn
    ntest <- 0
    depth[i]<-depth[i]-dh
### do while loop 2
    while(dh > 0){
      velo[i] <- discharge/topwidth[i]/depth[i]
      avgDepth <- 0.5*(depth[i-1]+depth[i])
      avgTopW <- 0.5*(topwidth[i-1]+topwidth[i])
      avgV <- discharge/avgTopW/avgDepth
      hydR <- avgDepth*avgTopW/(avgTopW+2.0*avgDepth)
      sFric <- (avgV^2)*(manningsN^2)/(hydR^(1.33))
      spDepth[i] <- spDepth[i-1]+(slopeChan-sFric)*dx
      spDepthT[i] <- depth[i]+(velo[i]^2/(2.0*9.81))
      difn <- spDepthT[i]-spDepth[i]
 #     print(difn)
#      print(cbind(i,depth[i],spDepth[i],spDepthT[i],dh))
        if(ntest > 0){
          dh <- dh/10.0
          break #break from do while loop 2
          }
        if(dif*difn > 0){
          break #break from do while loop 2
        }
      depth[i] <- depth[i] + dh
      ntest <- 1
    } #do while loop 2
  } # do while loop 1
} # for loop
##### report results #####
# build x-vector
distance <- seq(0,(imax-1)*dx,dx)
bottom <- -distance*slopeChan
watersurface <- depth+bottom
plot(distance,watersurface,ylim=c(0,max(watersurface)),type="l",col="blue",lwd=3)
lines(distance,bottom,col="brown",lwd=3)
```

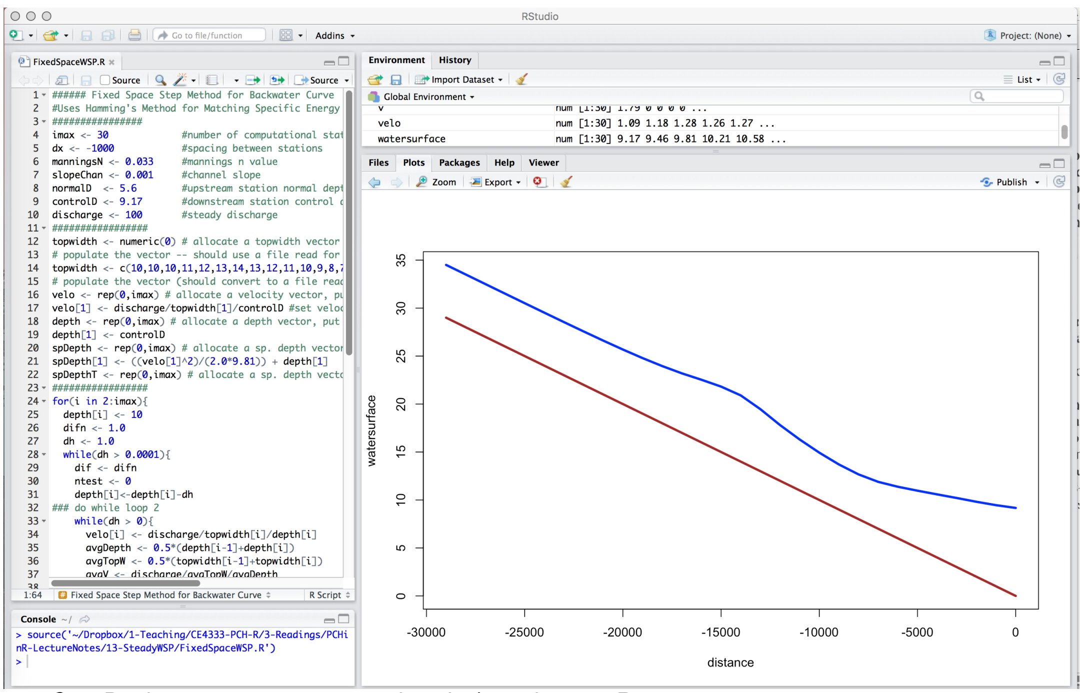


Figure 6. Backwater curve computed and plotted using **R**.

References

Koutitas, C.G. (1983). *Elements of Computational Hydraulics*. Pentech Press, London 138p. ISBN 0-7273-0503-4

Sturm T.W (2001). *Open Channel Hydraulics, 1ed.*. McGraw-Hill, New York.

Cunge, J.A., Holly, F.M., Verwey, A. (1980). Practical Aspects of Computational River Hydraulics. Pittman Publishing Inc. , Boston, MA. pp. 7-50