INCORPORATING SHARP-CRESTED WEIRS INTO IRRIGATION SCADA SYSTEMS

Tony L. Wahl[1]

ABSTRACT

Real-time flow measurement and monitoring are important components of modern irrigation SCADA systems. Many projects have existing sharp-crested weir structures that have not been incorporated into SCADA systems because they are partially contracted, and thus do not have a simple rating equation relationship. The Kindsvater-Carter procedure for calibrating partially contracted sharp-crested weirs is accurate and straightforward, but also somewhat tedious. Existing computer programs simplify the process but can presently be applied only to individual measurements. This paper presents a Microsoft® Excel spreadsheet model that can compute complete rating tables for sharp-crested weirs with full or partial flow contraction, using the Kindsvater-Carter procedure. Furthermore, through regression analysis, the spreadsheet determines a simplified rating equation that can easily be incorporated into remote terminal units (RTUs) and SCADA systems. The spreadsheet can be applied to fully contracted V-notch weirs with included angles of 25° to 100°, partially or fully contracted 90° V-notch weirs, partially or fully contracted rectangular weirs (including suppressed rectangular weirs), and fully contracted Cipoletti weirs.

INTRODUCTION

Real-time flow measurement and monitoring are important components of modern SCADA systems used on irrigation delivery and drainage projects. In open-channel applications, critical-depth flumes and weirs are common measurement devices. They are convenient for SCADA application, since discharge can be computed from a single water level measurement.

Long-throated flumes and broad-crested weirs are the recommended structure for most new installations, since they have the lowest head loss requirement of any critical-flow device and can be calibrated by computer analysis (Clemmens et al. 2001), but sharp-crested weirs are still commonly encountered on many irrigation systems. Rating tables and equations for fully contracted weirs are provided in many handbooks and manuals, but many real-world structures are not fully contracted, due to either inadequate flow depth or channel width upstream from the weir. These partially contracted structures have not been easy to incorporate into modern SCADA systems because their rating equations cannot be expressed

---

[1] Hydraulic Engineer, U.S. Department of the Interior, Bureau of Reclamation, Water Resources Research Laboratory, Mail Code D-8560, P.O. Box 25007, Denver, CO 80225-0007. E-mail: twahl@do.usbr.gov

in a simple form. Replacement of these structures with new devices having more reliable and accurate ratings and more tractable rating equations is not always practical or economically feasible.

Kindsvater and Carter (1957) produced charts for calibrating both fully and partially contracted sharp-crested weirs. These are recognized as the most accurate approach to sharp-crested weir calibration and have been incorporated into many manuals, guides, and international standards. Simple web-based tools exist for determining the flow corresponding to a single head (e.g., http://www.LMNOeng.com/Weirs/), but there has not been a convenient method for developing a simple and complete rating equation relationship that can be incorporated into a modern SCADA system. This technology gap has limited the integration of sharp-crested weirs into SCADA systems used for flow monitoring.

As a leader in western water management, the Water Resources Research Laboratory of the Bureau of Reclamation (Reclamation) works to develop tools that simplify and improve distribution, measurement and accounting for water on western water projects. This paper presents a Microsoft® Excel spreadsheet model that can be used to compute the rating tables for sharp-crested weirs with full or partial flow contraction. By regression analysis, the spreadsheet produces a simplified power-curve rating equation that can easily be incorporated into remote terminal units (RTUs) and SCADA systems. The spreadsheet can be used to analyze fully contracted V-notch weirs with included angles of 25° to 100°, partially or fully contracted 90° V-notch weirs, partially or fully contracted rectangular weirs (including suppressed rectangular weirs), and fully contracted Cipoletti weirs. Except for Cipoletti weirs, the algorithms used are those developed by Kindsvater and Carter (1957), which are also documented in Reclamation's Water Measurement Manual (2001) and Bos (1989). For Cipoletti weirs, the algorithm used is that described in Bos (1989), which includes an adjustment for velocity of approach at the head measurement location.

SPREADSHEET CONSIDERATIONS

Spreadsheets are widely used for engineering calculations and data analysis. Computational models can be easily created and tested, and spreadsheets facilitate production and presentation of results in table and chart form. Microsoft® Excel also provides a full set of graphical user controls and the ability to write custom macros and subroutines in Visual Basic. The spreadsheet presented here takes advantage of all of these capabilities.

A disadvantage of spreadsheets for distribution of finished applications is the fact that formulas, subroutines, and formatting of output are modifiable by the end-user. This makes it possible (perhaps even probable) that over time a spreadsheet application could be inadvertently modified or damaged by the user or a third party reviewing a particular file. Fortunately, Excel provides tools for protecting spreadsheets so that underlying formulas and routines cannot be modified. The

protection process used for this spreadsheet application will be described and some comparisons made to other forms of engineering software development and distribution.

USE OF THE SPREADSHEET

The spreadsheet was developed in Microsoft® Excel version 2002, but should operate correctly in earlier versions of Excel back to at least that included in Microsoft® Office 97. The file provided to the user is initially named *USBRWeir.xls*. Because the spreadsheet contains macros that must be executed as the user enters and changes data in the sheet, the user should set the macro security setting to Medium or Low, through the Tools|Options menu, prior to loading the spreadsheet file. The macros in the file are not digitally signed. Figure 1 shows a typical view of the spreadsheet as seen on a portrait-view screen.

Users should begin by personalizing their copy of the spreadsheet with their name and the location and description of the weir they are analyzing. The date field is automatically filled by the computer.

The weir type is selected from a list box, with five choices offered:

- 90° V-notch weir
- Rectangular suppressed weir
- Rectangular contracted weir
- Cipoletti weir
- V-notch weir (25° to 100°)

The first three weirs can be analyzed in either a partially contracted or fully contracted approach flow condition. The Cipoletti weir and the non-90° V-notch weirs must be fully contracted.

To the right of the weir type list box, the user can select either a manual or automatic recalculation mode. In both modes, whenever the characteristics of the weir and its approach channel are changed, the Dimension Check area of the page will display a new message indicating whether the weir as presently defined can be calibrated, and whether it is fully contracted or partially contracted. If the weir cannot be calibrated or is only partially contracted, suggestions are given for modifying the properties of the weir to enable its calibration, or to make it fully contracted (although many partially contracted weirs can be calibrated). In the automatic recalculation mode, the rating table will be fully recomputed after each change in the weir properties. In the manual mode, the user must click the Recalculate Rating button.

To the left of the Dimension Check area, the user can select preferred units of measurement and enter the dimensions of the weir and approach channel. The

rating table range can also be entered, and the desired head measurement resolution can be selected using a set of spin buttons. Resolution can be either 0.01 or 0.001 ft, or similarly 0.01 or 0.001 m.

The rating table is presented in a compact format that shows discharges at the intersection of head values indicated by the labels in the left-hand column and top row of the table. Below the rating table, a power-curve rating equation is shown

Figure 1. Example of Weir Calculator Spreadsheet.

4

XYZ Canal, Turnout 1

$$y = 2.4826x^{2.4815}$$

Figure 2. Example Rating Curve Output.
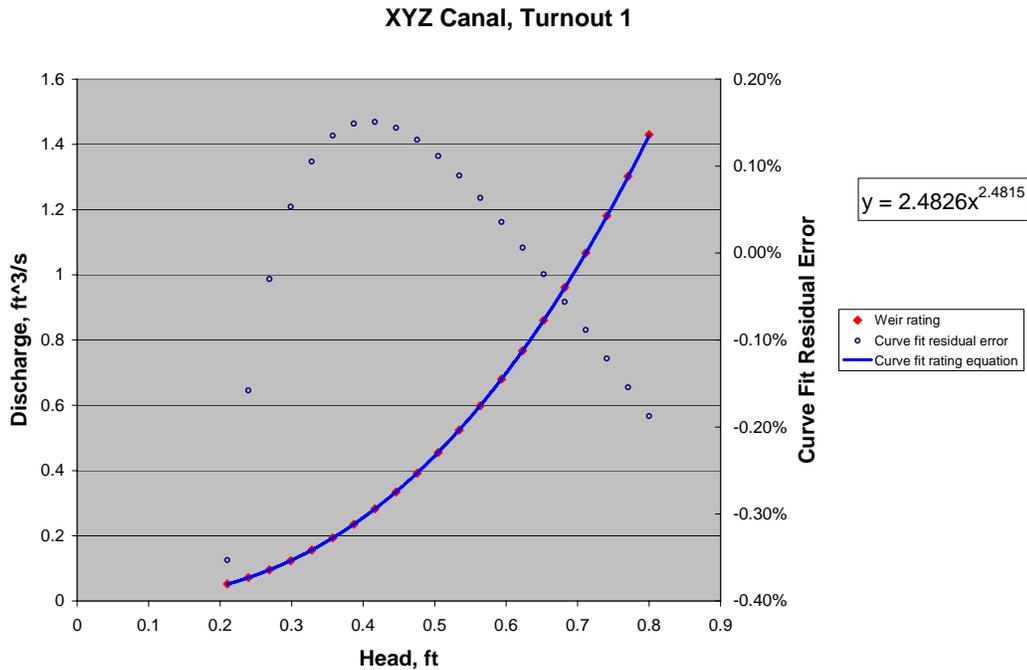
that can be used to compute flow from measured upstream head. Buttons to the right of the Weir Type list box can be used to print the rating table and head-discharge curve (Fig. 2).

SPREADSHEET DESIGN

Although the distributed (protected) version of the spreadsheet shows only one worksheet page to the user, the actual structure of the application is multifaceted. Seven worksheet pages exist in the file, most of them hidden from the user and protected against modification.

- **Input** – Always visible to the user. This sheet contains the graphical interface controls used to process input and display output.

- **Weir Rating** – Essentially a duplicate of the Input sheet without data input controls. The format of this page is optimized for its appearance when printed.

- **Curve** – Contains the head-discharge rating curve that can be printed by the user.

- **Weir Calcs** – Performs weir calculations for a single value of the upstream head.

- **Fit** – Organizes data for regression analysis to determine curve-fit rating equation.

- **Config** – This sheet contains information on configuration choices made by the user, e.g., units system, weir type, etc., and also provides calculations, logic, and temporary storage for results of the dimension check before it gets copied to the Input and Weir Rating sheets.

- **Charts** – This sheet contains the charts that define the Kindsvater-Carter weir formulas and other associated relationships. Regression analyses to convert the charts to computational form are contained on this sheet.

Calculations

The Weir Calcs sheet contains the majority of the actual hydraulic calculations. The sheet is set up so that it actually calculates the flow for all five weir types for a single input head provided in one cell of the worksheet; however, only one of the results is actually relevant at any given time. A Visual Basic subroutine manages the process of performing calculations over a range of heads to generate a complete rating table. The routine iteratively copies different values of upstream head into the input cell and retrieves the appropriate computed discharge from the Weir Calcs sheet. This result is copied into the rating table on the Weir Rating page. Similarly, the routine generates head-discharge pairs that are entered into the Fit sheet, where Excel's LINEST worksheet function is used to determine regression parameters for a linear function fitted to the log-transformed head-discharge data. The iterative copy and paste method of generating head-discharge pairs required a significant amount of Visual Basic programming, but is fast and efficient in operation. Other approaches, such as the use of Excel's two-cell Data Table command, seemed at first to be more elegant and require less programming, but in practice provided much slower performance.

The development and testing of software is always an interactive process, especially during the debugging phase; this is especially true in a spreadsheet application, where calculations and logic can reside in both worksheet cells and Visual Basic code modules. The fact that code modules can easily modify cell contents makes it very easy for errant code to have destructive consequences for the application. Thus, it is crucially important for the developer to save their work often and to archive the application at significant milestones.

Code Documentation

Building this application in a spreadsheet environment made it easy to document some aspects of the code and logic as the application was created. This was especially true on the Charts sheet, where data tables and charts that define the Kindsvater-Carter procedure are displayed, and the regression analyses used to convert the data tables to programmable equations are provided with each chart. The LINEST function was used to generate 3rd and 4th order polynomial regression equations to model most of the charts. In general, the regression equations reproduced the original charts with sufficient accuracy to avoid

introducing any significant biases through the modeling process. Many people use the regression tools in the Excel charting module (i.e., trend lines) for curve fitting, but the LINEST function is convenient because it provides more detailed output and can be replicated into multiple worksheet cells when a number of similar regressions need to be performed on different data sets. An example is the determination of equations for the C_e coefficient for partially contracted $90°$ V-notch weirs. Regressions were performed to define a family of C_e curves at different values of the ratio p_1/B_1 (crest height to approach channel width). It was then necessary to interpolate between adjacent curves, which was accomplished with formulas entered into worksheet cells.

It should be noted that the Excel LINEST worksheet function occasionally malfunctions when fitting 3rd and 4th order polynomials, a fact that has been previously observed by the author. The condition triggering this bug is not entirely clear, and the problem is not even always repeatable. The problem can be observed by plotting the resulting regression line onto a chart containing the original data. It is interesting to note that when the LINEST function fails, the Excel chart module still seems to draw accurate trend lines using a polynomial of the same order, and if requested to do so, it will print the correct equation coefficients onto the chart. Unfortunately, it does so in a fixed notation that shows only four digits to the right of the decimal, which is insufficient precision in many cases. This problem was encountered on a couple of occasions during the development of this spreadsheet. To solve the problem, data sets that exhibited this problem were manually copied to Lotus 1-2-3 version 5.0 for development of the regression equations. The equation coefficients were then manually transferred back into Excel.

Some aspects of code documentation are quite difficult to achieve in a spreadsheet application. Because formulas entered into worksheet cells often use cell references rather than variable names, deciphering them at a later date can be tedious. This problem can be overcome to some degree by generous use of range names for cells that contain specific data, such as gravitational constants, weir dimensions, etc.

SPREADSHEET PROTECTION

The spreadsheet makes use of four layers of security protection provided in Excel for packaging and protecting spreadsheets for public distribution:

- Hiding worksheets
- Protecting worksheets
- Protecting the workbook structure
- Locking Visual Basic projects

Hiding individual worksheets is accomplished with the Format|Sheet|Hide command on the Excel top menu. A similar Unhide command makes worksheets

visible again. Hiding worksheets removes visual distractions that the user does not need to be aware of, and prevents inadvertent modification of all data and formulas on the hidden sheets. Hidden sheets cannot be accessed by macros or subroutines, so it becomes necessary for the Visual Basic code itself to unhide and hide worksheets whenever access is needed to a hidden sheet. Hidden worksheets are not password protected and can be unhidden by the user unless the complete workbook is protected. Thus, by itself, the hiding of worksheets provides minimal security.

Protecting individual worksheets, using the Tools|Protection command, leaves sheets visible to the user (unless separately hidden as described above), but cell contents cannot be modified except for specific cells that are designated "Unprotected" by the worksheet designer. Again, protection also prevents Visual Basic code access, so macros and subroutines must unprotect worksheets before modifying them, and then protection must be reapplied internally by the code. Passwords can be assigned when protection is applied, but since subroutines in the Visual Basic code must be able to unprotect and protect sheets, the password must be embedded in the code, where it could be seen by the user. Thus, worksheet protection also offers minimal security by itself.

Protecting the entire workbook structure prevents the user from inserting, deleting, hiding, showing, moving, or renaming individual worksheets. The combination of hiding sheets and protecting workbook structure creates an additional layer of protection, but again it is sometimes necessary for the Visual Basic code to have access to hidden sheets. Thus, the code must be able to temporarily remove the workbook structure protection in order to reveal worksheets, and for this purpose the password used to apply and remove workbook structure protection must be embedded in the code.

The final item required to protect the application is to prevent the user from viewing the Visual Basic code, since it contains the passwords for hiding and protecting the worksheets and workbook, as well as the logic of the application. Inadvertent modification of the code could cause the worksheet to stop functioning or produce erroneous results. The Visual Basic code can be hidden and locked from the Visual Basic Editor by choosing the Tools|VBA Project Properties command. On the Protection tab of this dialog box, select Lock Project for Viewing and assign a password.

The protection process is analogous to the compilation of source code to create a stand-alone executable application. An unprotected version of the worksheet (i.e., the source code) should be maintained by the designer in parallel with the protected, distributed version (i.e., the compiled application). This saves one from having to go through the process of unprotecting the distributed version when modifications or improvements need to be made, and also provides insurance against losing or forgetting the critical password for gaining access to the Visual Basic code.

```
Public Sub ProtectApplication()
  Dim s As Object
  For Each s In Application.Sheets
    Call ProtectSheet(s.Name)
    If s.Name <> "Input" Then Call HideSheet(s.Name)
  Next
  Call ProtectWorkbook
End Sub

Public Sub ShowSheet(Sheetname As String)
  Call UnprotectWorkbook
  Sheets(Sheetname).Visible = True
  Call ProtectWorkbook
End Sub

Public Sub UnprotectSheet(Sheetname As String)
  Sheets(Sheetname).Select
  ActiveSheet.Unprotect Password:="sheetpassword"
End Sub

Private Sub UnprotectWorkbook()
  ActiveWorkbook.Unprotect Password:="bookpassword"
End Sub

Public Sub HideSheet(Sheetname As String)
  'Comment out all lines of this subroutine in unprotected version
  'Call UnprotectWorkbook
  'Sheets(Sheetname).Visible = False
  'Call ProtectWorkbook
End Sub

Public Sub ProtectSheet(Sheetname As String)
  'Comment out all lines of this subroutine in unprotected version
  'Sheets(Sheetname).Select
  'ActiveSheet.Protect Password:="sheetpassword", DrawingObjects:=True,
       Contents:=True, Scenarios:=True
End Sub

Private Sub ProtectWorkbook()
  'Comment out all lines of this subroutine in unprotected version
  'ActiveWorkbook.Protect Password:="bookpassword"
End Sub
```

Figure 3. Key Visual Basic Subroutines for Protecting the Spreadsheet.

The protection process is somewhat tedious and must be repeated whenever changes are made to the application. To simplify the work, the subroutines that hide and protect worksheets are collected together in one Visual Basic module, as shown in Figure 3. All lines of code shown in the last three subroutines (HideSheet, ProtectSheet, and ProtectWorkbook) are commented out in the unprotected version. When one wishes to create a new protected version, four steps are required:

1. Remove comment marks (') to enable the code in the last three subroutines in Figure 3 (HideSheet, ProtectSheet, and ProtectWorkbook)

2. From the Visual Basic Editor, choose Tools|VBA Project Properties. Go to the Protection tab and check "Lock Project for Viewing", and assign a password.

3. Run the ProtectApplication macro, which will automatically:
   - Protect each individual worksheet with a first password
   - Hide all worksheets, except the main sheet, named "Input"
   - Protect the entire workbook with a second password

4. Save the protected application under a new name, e.g., *ProtectedFile.xls*

CONCLUSIONS

The spreadsheet application documented here is expected to be a useful tool for incorporating sharp-crested weirs into SCADA systems used on irrigation and other water resources projects. The application will enable users to develop detailed rating tables and simplified rating equations for the most common types of partially and fully contracted weirs. The spreadsheet checks weir and approach channel dimensions to verify that ratings can be determined, and then applies the widely accepted Kindsvater-Carter weir relationships to develop a head-discharge rating table. Regression analysis techniques are then used to obtain a simple power-curve equation relating discharge to measured upstream head.

Although spreadsheet applications are not compiled, the application has been secured through a multi-tiered approach described in this paper, so users can be confident that the distributed spreadsheet has not been modified from its original form. One area that may be worthy of some future investigation is the digital signing of the Visual Basic macros and subroutines that accompany the spreadsheet. This would potentially free the user from the need to enable macros each time the spreadsheet is loaded. However, initial research showed that the steps required to produce digitally signed macros are complex and poorly documented, and at this time, few organizations are making use of this security technology.

The weir calculator spreadsheet is available at this time in a beta-test version from the web site of Reclamation's Water Resources Research Laboratory. The URL is **http://www.usbr.gov/pmts/hydraulics_lab/usbrweir/**.

REFERENCES

Bos, M.G., 1989. *Discharge Measurement Structures*, ILRI Publication 20, International Institute for Land Reclamation and Improvement, Wageningen, The Netherlands, 401 p.

Bureau of Reclamation, 2001. *Water Measurement Manual*, 3rd ed., revised reprint, U.S. Government Printing Office, Washington, DC 20402.

Kindsvater, C.E. and R.W.C. Carter. 1957. Discharge Characteristics of Rectangular Thin-Plate Weirs. *Journal of the Hydraulics Division*, Proceedings of the American Society of Civil Engineers, Vol. 83, No. HY6. Paper 1453.

Clemmens, A.J., T.L. Wahl, M.G. Bos, and J.A. Replogle, 2001. *Water Measurement with Flumes and Weirs*, ILRI Publication 58, International Institute for Land Reclamation and Improvement, Wageningen, The Netherlands, 382 p.