

A computer program for stream rating curves

Alternative Hydraulics Paper 9

John D. Fenton

Institute of Hydraulic Engineering and Water Resources Management
Vienna University of Technology, Karlsplatz 13/222,
1040 Vienna, Austria
<http://johndfenton.com/>
<mailto:JohnDFenton@gmail.com>

Saturday 27th February, 2021

This document describes the use of a computer program that reads in rating data for a gauging station or structure and calculates a rating curve using least squares methods. It is based on the author's research described in two documents, Fenton (2015), a technical report, and Fenton (2018), a journal article. They describe the nature of rating data and curves, the problems of obtaining rating curves from data, advocating the use of two different forms of approximation, polynomial and spline, and applying them to several problems, including allowance for the age of data points, the computation of historical curves, and the concept of an envelope to the data. The program "Rating-curve" described here uses spline approximation to solve just the classical problem, to fit a continuous curve to a number of rating points.

All files necessary for the operation of the program are in Program Files, a self-extracting file that, when downloaded and executed (after your computer asks you to say that it is acceptable), unpacks the files, retaining the original file structure, under a directory of your choice.

This report is: Fenton, J. D. (2021) A computer program for stream rating curves, Alternative Hydraulics Paper 9. <http://johndfenton.com/Rating-curves/Instructions.pdf>

Contents

1	Introduction	2
2	Method	3
3	Obtaining and using the program files	6
4	Output and plotting of rating curves	8
5	Results	9
6	Scattered, looped, and historical data	10
7	Conclusions	10

1 Introduction

A rating curve is a relationship between the discharge Q of a stream and h , the stage or surface elevation at a gauging station or a structure. The curve is calculated from a number of (h, Q) rating data points from that station, using relatively infrequent measurements of the velocity distribution, cross-section, and stage of the stream. Then later, when routine measurements of stage are made, the flow can be estimated from the rating curve.

Traditional methods for the calculation of stream rating curves are mistaken and outdated. They are based on a belief that data points should lie on a straight line when plotted on a graph with axes $\log Q$ and $\log(h - h_0)$, where h_0 is a value that has to be determined. Such a relationship appears ubiquitously in books, standards, and university lecture notes. Whereas it is sometimes a convenient approximation to the relationship $Q(h)$ over the whole range of data, in general it is not. It is certainly not expected to hold in a piecewise sense, which apparently is a widespread belief. It is an unnecessary complication, while being an over-simplification of the real hydraulics at many gauging stations. Such a formula is valid for an infinitely-wide weir in infinitely-deep water or for uniform flow in an infinitely-wide rectangular channel. There is no reason for a real rating curve to follow such a function closely. Insufficient knowledge of hydraulics has led to a misplaced belief in such a function. This has led to complicated procedures in some organisations where a great deal of trouble goes into the laborious manual fitting of piecewise-continuous straight lines on logarithmic axes by adjusting the offsets h_0 for each on different vertical $\log(h - h_0)$ axes.

Fenton (2015, 2018) used least-squares approximation for the automatic computation of rating curves. Two methods were considered.

Global polynomial approximation was discussed theoretically, revealing some dangers that may make it fragile and unreliable, which might explain the lack of adoption by the profession. A computationally-robust method using Chebyshev polynomials was then developed which overcame most such problems. This was tested extensively by McMahan and Peel (2019) to obtain 622 rating curves from 171 Australian Bureau of Meteorology Hydrologic Reference Stations. They found that the method worked well – with the exception of about 0.5% of the stations, where there was difficulty approximating the low-flow data.

A problem with such polynomial approximation is that initially one does not know what the degree of the polynomial should be. As the degree increases, gradually better agreement with the data points is found, however at some stage the degree of approximation becomes too high and unacceptable oscillations appear. That was not a problem for Fenton or McMahan and Peel, where the appropriate degree in each case was found by monitoring and trial and error. Indeed, once the characteristics of the data for any particular station have been established by monitoring, the polynomial degree can be used for all future calculations as more rating points are added. However, for the development of a stand-alone computer program such as the subject of this document, it is not quite good enough.

Approximating splines was the second method developed, and which overcame the remaining problems of polynomial approximation. The data is approximated in a piecewise sense, using a sequence of second-degree polynomials. Whereas using global polynomials the approximation can fail spectacularly (and does every time if a too high degree of approximation is used) using splines the approximation is rather more reliable, and at worst, if a problem is not sufficiently accurately approximated, the problem can be remedied by a judicious choice of stage knot points. Whereas this still requires initial monitoring of the results in each case, the whole method is more flexible and reliable.

With this in mind, the author decided that it would be practicable to release a working version of the spline program for public use, which this document describes.

Both global polynomial and spline approximation methods allow the specification of a weight for each data point, enabling the filtering of data, possibly incorporating its age, and allowing the computation of

a rating curve for any day in the past up to the present (Fenton, 2015, 2018). However that has, as yet, not been included in the present computer program.

The author would welcome any criticism, feedback or suggestions, at johndfenton@gmail.com.

2 Method

This section provides some theoretical background to the program.

2.1 Use of Q^v as the quantity to be approximated

Fenton and Keller (2001, §6.3.2) suggested that there are advantages in writing a polynomial approximation for Q raised to the power v , specified *a priori*, i.e.: $Q^v = f(h)$, where $f(h)$ is the function to be determined. They recommended a value of $v = \frac{1}{2}$, on the basis of that being the mean value in the hydraulic discharge formulae for a sequence of weir and channel cross-sections that modelled local and channel control (Fenton, 2001). The use of such a fractional value of v has two effects:

1. For small flows, with h and Q small, the data usually is such that

$$Q^v = a_0 + a_1 h, \quad (1)$$

where $v = \frac{1}{2}$, is a surprisingly good approximation when compared with power function approximations in which v is a free parameter, as shown in Fenton (2015, §3.4). We note that on the above-scorned $(\log Q, \log(h - h_0))$ axes, equation (1) would indeed plot as a straight line. In this small flow limit more general polynomial and spline approximations just have to simulate nearly-linear variation, which, in particular splines, can easily do.

2. For larger flows, when variation with h is more general, the use of Q^v with a fractional v , maybe $\frac{1}{2}$, means that the magnitude of the dependent variable to be approximated is numerically much smaller, so that, instead of a range of say, $Q \approx 1$ to $10^4 \text{ m}^3 \text{ s}^{-1}$, for $v = \frac{1}{2}$ a numerical range $Q^{1/2} \approx 1$ to 10^2 has to be approximated. This is a simple version of a *power transformation* used in more formal data analysis applications to stabilise variance and to make the data more normal distribution-like.

The experience of the author is that using splines, as implemented in the accompanying program, the value of $v = \frac{1}{2}$ is always good enough and it is not necessary to use any of several methods outlined in Fenton (2015, §3.2) to calculate a more general value. If, for any gauging station, problems were found, these would be reduced by using more spline knot points at the low flow end.

2.2 Approximating splines

Using piecewise-continuous splines, the parameters of the problem are the number and stage values of knot points between which simple quadratic or cubic spline functions are used. This does not mean that the knot points are interpolation or approximation points – only the values of stage are specified.

A simple automatic method for the placing of those knots is just to require that there be the same number of data points in each interval between knots. This usually works well. Otherwise, in difficult cases the values of stage for the knot points could be specified, such as in regions of rapid variation.

In the usual application of splines, they interpolate all data points; here they are required to perform least-squares approximation. The method is in a sense both local and partly-global, in that a sequence of local polynomials is used, each of which approximates just part of the range of data, but which is required to merge smoothly with its neighbours, so that they and their approximated data points also have some

influence. It seems to be relatively simple, to have a readily-understandable physical significance and a good level of continuity. Figure 1 shows the principle behind spline approximation.

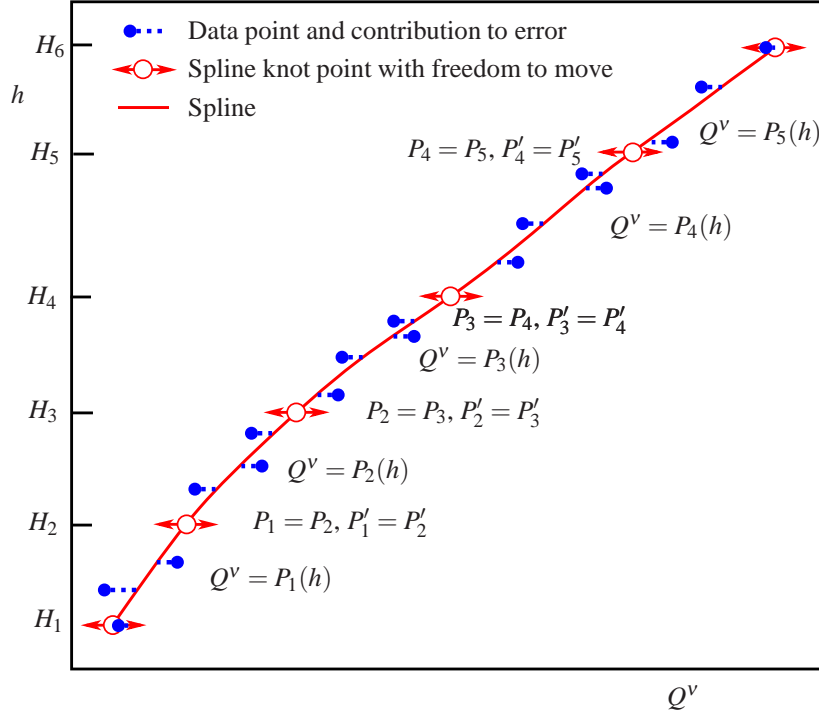


Figure 1: Piecewise-continuous second-degree spline approximation of rating points, showing the ability of knot points to move so as to minimise the total error, while maintaining continuity of function and slope across the knot point

Consider a number of data pairs (h_n, Q_n^v) for $n = 1, 2, \dots, N$. Let the range of stages $[h_{\min}, h_{\max}]$ be subdivided into J intervals by $J + 1$ knot points, at each of which the stage is H_j , $j = 1 \dots J + 1$ and $H_0 = h_{\min}$ and $H_J = h_{\max}$. They must be separate and ordered, such that $H_{j+1} > H_j$ for all the j . The points are to be approximated over each interval by a polynomial of degree $M = 2$ or 3 . So, for $j = 1, \dots, J - 1$ and $H_j \leq h \leq H_{j+1}$:

$$P_j(h - H_j) = \sum_{m=0}^M c_{j,m} (h - H_j)^m, \quad (2)$$

such that $P_j(h - H_j)$ is the polynomial of degree M which holds between stage knot points H_j and H_{j+1} , expressed as a function of $h - H_j$, the height of stage h above the preceding knot point.

At each knot there are continuity conditions between the polynomials on either side, agreeing in value and first derivative for quadratic splines, and additionally the second derivative for cubic splines. The H_j may be set as data or automatically allocated such that there are the same number of data points in each interval, for example.

The program calculates all the coefficients so that the sum of the squares of the differences between data points and functions are minimised. Only those data points falling within a certain interval contribute to the total sum using the function for that interval, as given in Eqn (2), such that

$$\epsilon_2 = \sum_{j=1}^J \sum_{n \in I_j} w_n \left(\sum_{m=0}^M c_{j,m} (h_n - H_j)^m - Q_n^v \right)^2, \quad (3)$$

where I_j is the set of data point reference numbers in interval j ; the summations are a mathematical statement of the obvious “over all the intervals, consider every data point in that interval and calculate its contribution to the total error using the polynomial for that interval”.

The spline nature of the approximation requires the satisfaction across each interior knot of the continuity of function value plus all derivatives up to $M - 1$ at the knot points, giving extra conditions on all the

coefficients $c_{j,m}$. From equations (2) at left and right of each interior knot point, and using $\delta_j = H_{j+1} - H_j$ for the interval length, the conditions become, for $j = 2 \dots J$:

$$c_{j+1,0} = \sum_{m=0}^M c_{j,m} \delta_j^m, \quad c_{j+1,1} = \sum_{m=1}^M m c_{j,m} \delta_j^{m-1}, \quad \text{and}$$

$$c_{j+1,2} = c_{j,2} + 3 c_{j,3} \delta_j, \quad \text{if } M = 3. \quad (4)$$

The solution procedure to determine the unknown $c_{j,m}$ by minimising ε_2 in Eqn (3) is, first to use the continuity equations (4) to eliminate those variables on the left sides, then to solve for the remaining $c_{j,m}$ by minimising the value of ε_2 . Because of the high degree of continuity enforced by Eqns (4), the net number of unknowns is not large, just $J + M$. In the present work, quadratic splines $M = 2$ was found to be best, and J might have a maximum value of, say, 10.

To allocate the knot points H_j , they could simply be equally spaced between minimum and maximum values. Another automatic method, that which was used here, was simply programmed and which often worked well was where knot points were calculated such that there were approximately the same number of data points in all intervals. However, the knots can also simply be assigned as data. Such hand adjustment of the knot positions was sometimes useful to describe regions of high curvature. It was usually found necessary to place a knot point near isolated data points for extreme floods. It should be emphasised that only the values of the stage knot points H_j were specified, there was no attempt to pin the approximation by specifying the discharge there.

The approximating spline method worked well in all the examples considered, with occasional improvement by hand allocation of knot points.

The program was written in the C language, and uses specialist software provided in Press et al. (1992), including routines sort2 to sort the input, and powell and subsidiary routines to perform the optimisation.

3 Obtaining and using the program files

All files are in <http://johndfenton.com/Rating-curves/Program-Files.exe>, which is a self-extracting file that contains the original directory structure. The base directory to which the files are extracted can be anywhere in the file system.

Table 1: File structure in the self-extracting file (directories are shown bold). The directory **Data** and sub-directories are not strictly necessary. They give some examples to test the running of the program. As explained on the next page the program can approximate data stored anywhere on the computer, it just needs to be instructed.

Your-directory-name		
Instructions.pdf		This document.
Rating-curve.exe		The program, can be run by usual methods.
Control.dat		The first line of this tells which directory and which data files to use. Will have to be edited for every new data set. All lines after the first, probably reflecting the user's history, are ignored.
Settings.dat		These should not have to be changed.
Plot.plt		A Gnuplot file. After software for that is installed and allocated to *.plt, can be run by double clicking in a file browser, or run in a Gnuplot screen with changing directory to this one and opening it.
Source.plt		A file created each run, telling Gnuplot where to look for data.
Run-and-plot.bat		Running this executes the program, pauses to see output summary, and then runs Gnuplot. It will have to be edited to give a program location.
Data		
Mystery	Data-with-no-knots.dat	An everyday example, with no knots given.
Australia	Shannon.dat	A case that was difficult using polynomial methods, but is not a problem for splines – with the knots provided.
	Shannon.knots	
Mirza	Ganges92.dat	Relatively few data points. Using 10 intervals might be overfitting, but it seems to work well.
	Ganges92.knots	
USGS	Noxubee_70s.dat	Illustrating the power of the method, without messing around with various straight lines on log-log axes.
	Noxubee_70s.knots	

Fitting a rating curve to data can now be done very simply: in the first line of the file Control.dat provide the name of the directory “C:/.../Mydirectory” and the name of the data set “Mydata” separated by a space or tab. Then in C:/.../Mydirectory provide a file Mydata.dat which, after a header line for identification purposes, contains two columns of (possibly unsorted) stage and discharge values.

After that you might care to tweak the program using a knots file as described below.

Control.dat – the program reads just the first line which contains (Directory without “/” and Filename without extender).

Data/Mystery	Data-with-no-knots
Data/Australia	Shannon
Data/Mirza	Brahmaputra92
Data/Mirza	Ganges92
Data/USGS	Noxubee_70s
C:/Your-full-directory-path-without-a-terminating-slash	Filename
../Your-relative-pathname without slash	Filename

Filename.dat contains the rating data. It has the following structure – any lines that are not blank or data must start with #, which is a comment for the plotting program Gnuplot which is recommended:

```
# Brief description/label of the data set or site that will be printed out and shown on figures
Zero or more blank lines
Then the rating points: on each line a data pair in the form “Stage Discharge”, separated by
a space or tab, then possibly other information, which will be ignored. The program senses
the end of file itself. The data can be in any order, the program sorts them according to
Stage.
```

As far as units are concerned, the program doesn’t care or notice if you are in Liberia, Myanmar or the USA where you can use the customary Imperial units, or elsewhere where *S.I.* units are usual, or even in the author’s native Australia where discharge might be in terms of the quaint Megalitre-per-day units.

Filename.knots – if you provide a knots file, it has the structure:

```
# A heading line for identification that is ignored
Zero or more blank lines
An integer number   If the number is positive, the program sets the number of inter-
                    vals to this, and the knot points are calculated automatically. Any
                    numbers below are ignored. If the number read is negative, its
                    numerical value is meaningless, and the program reads in the sub-
                    sequent interior spline knots stages, one on each line.

Zero or more blank lines
A number, with or without decimal point for the first interior knot
A second number
And so on.          The program stops reading when it finds the last number in the
                    file, which is the last interior spline knot. The program then sets
                    the first and last knots to be the minimum and maximum stages.
```

Settings.dat should not have to be changed

```
# Standard settings file. Could be changed if required, but probably not necessary
0.5  Value of v. Should not have to be varied
8    Default number of intervals in case no knots file is present
2    Degree of splines. Could be 2 or 3. For rating curves 2 is better; 3 is too flexible
100  The number of rating point pairs output
2    Spacing in stage of points output: 1 linearly, 2 quadratically (decimal values possible)
```

4 Output and plotting of rating curves

The program produces, in the same directory as the data file `Yourdata.dat`, an ASCII file `Yourdata.res` containing two columns, of 100 (specifiable in `Settings.dat`) pairs of Stage and Discharge values, from the smallest to the highest stage, sorted according to stage, and spaced linearly or quadratically (much closer together at the bottom) also as specified in `Settings.dat`. These could then be read into a spreadsheet such as “Calc” in Libre Office <https://www.libreoffice.org/> (or any other spreadsheet or software that you might know :-)

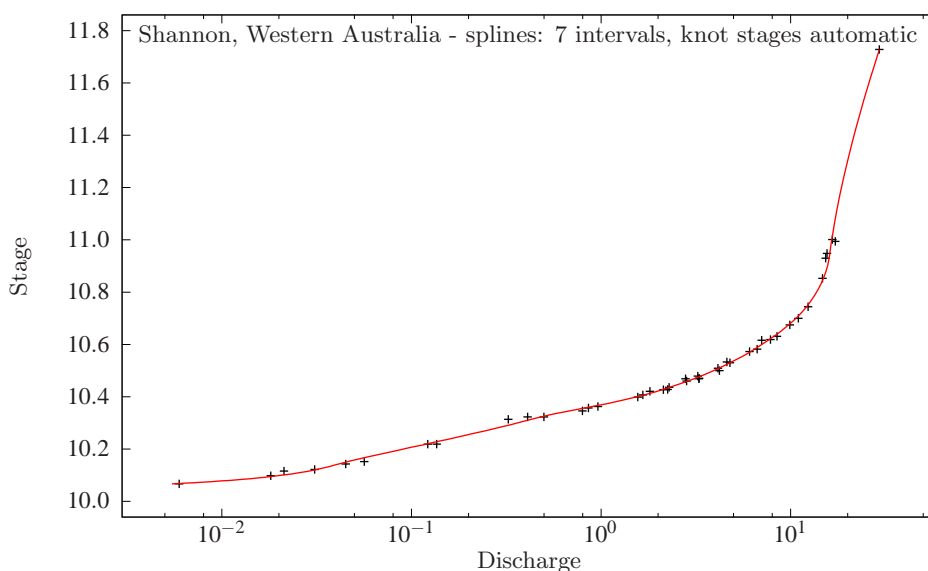


Figure 2: Typical figure produced by Gnuplot and script file `Plot.plt`

For plotting, the solution that the author prefers is the program Gnuplot <http://www.gnuplot.info/>. The present rating curve program, in addition to outputting the results to `Yourdata.res` outputs a file `Source.plt` which tells the Gnuplot file `Plot.plt` where to find the data file `Yourdata.dat` and to plot it and the results file `Yourdata.res`. One might have to instruct the computer to associate the file name extender “*.plt” with Gnuplot. After that, appropriate clicking on `Plot.plt` in the home directory of the program should produce a figure. The little batch file `Run-and-plot.bat` provided does just what it says, double clicking on it in a file browser runs the program and after pausing, plots the figure. That can then be copied, or saved as a graphics file or a PDF file. An example is shown in Figure 2, based on the site described in the Introduction that was troublesome for polynomial approximation. In fact, it was unusual in that it consisted of a compound V-notch weir. The present spline program managed the problem well – after some experimentation with knot placement.

As discharge can vary by some three or four orders of magnitude, for plotting it is recommended to use the traditional horizontal scale of $\log Q$. However, for stage the use of logarithmic scales has sometimes made representation and understanding difficult and caused errors. Fenton (2015, §4.3) reported on elementary mathematical mistakes in then-current software using logarithmic axes. Unless one were using the traditional power function with vertical axis $\log(h - h_0)$ beloved of traditionalists, the logarithm of stage has no meaning or utility and there is no reason to use it.

Usually a gauging station has an arbitrary local datum somewhere below the lowest possible water level, so that stages plotted are typically in the range of greater than 0 m to something like 10 m or rather less, but where they have no external physical significance. Using the present software it would be possible always to use the actual elevation above mean sea level, which would have certain advantages, as that is often important in water engineering. There would no longer be such a need to maintain an arbitrary local benchmark and refer measurements to that.

5 Results

To examine the performance of the methods described above, data from six sites are considered, two from Australia, two from Bangladesh, and two from the United States of America. Wherever necessary results were converted to *SI* units. Results are shown in Fig. 3. All results are for quadratic splines, $M = 2$. No advantage was found using cubic splines, $M = 3$. In the first four cases in Fig. 3, automatic allocation of knot points was used, with about the same number of data points in each interval; in the two remaining examples, especially at large flows where there were few data points, the knot points H_j were allocated manually by specifying in a data file. The method used is shown on each figure.

It can be seen that the figures represent a variety of behaviour and numbers of gaugings. Not the least problem is the wide gap for high flows in Figure 3a, yet the splines make a plausible bridge. The method can still work with few data points in an interval. The performance seems to be uniformly good.

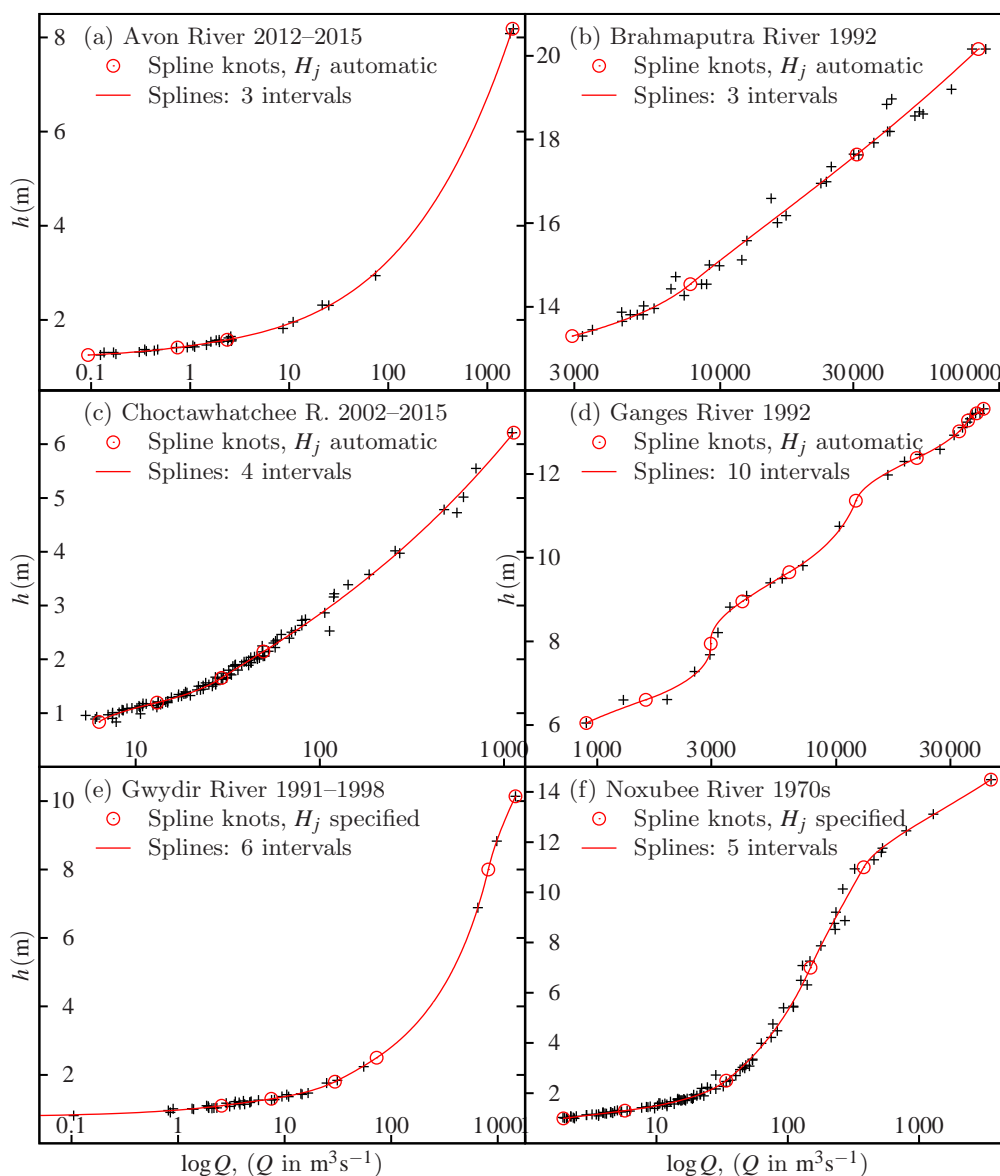


Figure 3: Typical rating curves obtained from second-degree spline approximations

Figure 3d shows a demanding problem, if one accepts that the fine structure (“lumps”) in the data are real. It has been chosen here to do that as a test of the model, although it might be over-fitting it. If

one knew the reliability of the data, it might have been possible to use a smoother approximation, with fewer intervals. The splines have performed well, even if the placement of the knots has yet again been performed automatically, and there are actually few data points in each interval.

Again, Figure 3e shows data with some large gaps. Using splines, automatic allocation of equally-populated intervals did not give such good results for high flows. So, for the first time in Figure 3, knots were allocated by hand, placing one at 8 m, which caused the splines to pass very close to the high flow points as shown. Some of those data points were obtained partly from a boat floating above flooded agricultural land — possibly a rougher approximation would have been more justified. In any case with splines one has quite a lot of power to experiment with point placement.

6 Scattered, looped, and historical data

The current program performs a relatively simple task. In Fenton (2015, §§10-12) and Fenton (2018, §6) the author considered several other problems which can be handled by rather more general programs. If one knows the age of every rating point it is relatively easy to generate rating curves for any day in history, and to show the movement of the rating curve at a station.

In the case of scattered data such as at highly alluvial sites, the present methods of approximation can be used to generate envelopes to the data, so that for a given stage, one can also obtain the maximum and minimum likely values.

With data approximation as used here, the incorporation of more recent and possibly outlying data is capable of being handled more rationally than the Procrustean use of straight lines on log-log axes. If a discrepant rating point is determined, there is no point in locally distorting the rating curve by using a shift curve, as the next time that the interval of validity of the shift is visited by a flow event, conditions in the stream might be quite different. This has been discussed at length in the two papers cited above.

7 Conclusions

The problem of rating curve generation benefits little from simple hydraulic formulae, and is essentially one of data approximation, for which least-squares methods can be used. They too have their problems if used without knowledge and overcoming of intrinsic computational difficulties. Using piecewise-continuous splines overcomes most of these problems. That can be done with no information other than the rating data and letting the program choose a certain number of knot points, or specifying that number. Sometimes specification and manipulation of knot point stage values might be necessary.

References

- Fenton, J. D. (2001), Rating curves: Part 2 - Representation and approximation, in *Proc. Conf. on Hydraulics in Civil Engng*, 28-30 Nov., Hobart, Instn Engrs, Aust., pp. 319–328. <http://johndfenton.com/Papers/Fenton01-Rating-curves-2-Representation-and-approximation.pdf>.
- Fenton, J. D. (2015), Generating stream rating information from data, Technical Report 8, Alternative Hydraulics. <http://johndfenton.com/Papers/08-Generating-stream-rating-information-from-data.pdf>.
- Fenton, J. D. (2018), On the generation of stream rating curves, *Journal of Hydrology* **564**, 748–757.

- Fenton, J. D. and Keller, R. J. (2001), The calculation of streamflow from measurements of stage, Technical Report 01/6, Cooperative Research Centre for Catchment Hydrology, Melbourne. <http://www.ewater.org.au/archive/crcch/archive/pubs/pdfs/technical200106.pdf>.
- McMahon, T. A. and Peel, M. C. (2019), Uncertainty in stage-discharge rating curves: application to Australian Hydrologic Reference Stations data, *Hydrological Sciences Journal* **64**(3), 255–275.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (1992), *Numerical Recipes in C*, second edn, Cambridge.