

Novel software developments for the automated post-processing of high volumes of velocity time-series



M.A. Jesson, J. Bridgeman*, M. Sterling

School of Civil Engineering, University of Birmingham, Birmingham B14 2TT, United Kingdom

ARTICLE INFO

Article history:

Available online 7 July 2015

Keywords:

Batch processing
Despiking
Acoustic Doppler velocimeter
Turbulence
Phase-space thresholding
Filtering
Software
Fluid mechanics

ABSTRACT

This paper describes novel software developments designed to automate the parsing, filtering (despiking) and calculation of mean flow and turbulence parameters of velocity time-series. The software was written to facilitate the processing of the large number of time-series (approximately 4000) generated by the authors' experimental work, mapping velocity and turbulence fields in a laboratory flume using an acoustic Doppler velocimeter. The software will import all, or selected, data files from a specified directory, desprike them using one of a number of filters (Phase-Space Thresholding, for example) and graphically present the parameter distribution over the cross-section under investigation. Many different data file formats are supported, including Nortek binary formats, Turbulent Flow Instrumentation "Cobra" probe format and delimited plain-text files, meaning that the software has applications in a significant number of different environmental engineering applications where large data sets are routinely collected and analysed.

© 2015 The Authors. Published by Civil-Comp Ltd. and Elsevier Ltd.

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The purpose of this paper is to describe software, the velocity signal analyser, ("VSA") developed to automate the parsing of data files of velocity time-series, and the post-processing of the data. It is anticipated that the VSA software will prove to be of great value to other researchers investigating velocity and turbulence fields. Before describing the software fully, a brief overview of the work which served as the motivation behind the development of the software is given as this will serve to clarify some of the functionality it provides. This paper is based on Jesson et al. [1], but the current paper includes additional description of time series analysis and, in particular, the quadrant-hole analysis functionality for identification of turbulence propagating events.

The modelling of river flow is important for a number of reasons, from predicting flood events to evaluating the effects of proposed river engineering works to understanding scouring processes. In order to investigate open-channel flow over heterogeneously rough beds the authors undertook a series of experiments using a rectangular flume in the University of Birmingham Civil Engineering Laboratories. The aim of this work (which is described fully in [2–4]) was to identify distributions of the 3-D velocity components, Reynolds stresses and other turbulence parameters, and also to identify coherent structures with the flow. The data was used to calculate the

parameters of the Shiono–Knight method (SKM; see [5,6] for details), a recognised open–open channel flow model, allowing recommendations to be made regarding its application in rivers with heterogeneous bed roughness.

This experimental work involved creating a high resolution map of the velocity and turbulence distributions over cross-sections of the flume. Two bed configurations were used, one with full-length, longitudinal strip roughness (smooth PVC on one side of the channel and rough gravel on the other) and the other with a "checkerboard" of rough and smooth sections (illustrated in Fig. 1). Each permutation of bed configuration, discharge and longitudinal position of the measured cross-section resulted in a new data-set consisting of over 500 individual data-points, each situated at the node of a 10 mm spaced grid spanning the cross-section. Here, data-point refers to one location at which a velocity time-series has been recorded; a data-set consists of a number of data-points. Six full data-sets were gathered, along with six partial data-sets, making a total of approximately 4000 data-points. A 60 s long, 200 Hz, 3-D velocity measurement was made at each data-point, resulting in 12,000 instantaneous measurements in each dimension per data-point.

The measurements were made using Nortek acoustic Doppler velocimeters (ADV). ADVs transmit pulses of high-frequency sound waves which are reflected back to the ADV by impurities in the water. The frequency of the reflected waves is Doppler shifted due to the velocity of the impurities, allowing calculation of the 3-D velocity. ADV data are subject to invalid measurements (caused by aliasing when the magnitude of the phase shift is greater than 180° and also

* Corresponding author. Tel: +44 121 414 5145; fax: +44 121 414 3675.
E-mail address: j.bridgeman@bham.ac.uk (J. Bridgeman).

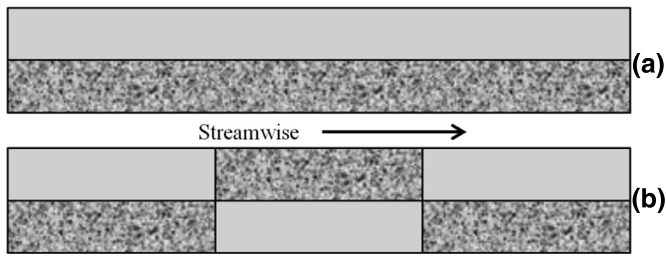


Fig. 1. Schematic of the heterogeneous channel bed configurations: (a) full-length, longitudinal strip roughness; (b) checkerboard roughness.

insufficient impurities to reflect the sound waves – see [2,7,8] which appear in the form of “spikes” in the time-series. These spikes may significantly affect turbulence parameters calculated from the time-series and thus should be removed from the time-series in a process termed “despiking” (see [8] for an overview of the causes of these spikes and the despiking process).

In common with many instruments, the raw data from the ADV is recorded in a proprietary, binary format. While this has the advantage of reducing file size, it necessitates a somewhat convoluted process to extract this data before any post-processing or analysis can be performed. Streamlining this process was important in order to reduce data analysis times to reasonable levels – as an example, it takes approximately 10 min to import and despike one of the authors’ datasets (530 Nortek Polysync files, making 412 MB of raw data) on a laptop with a 2 GHz Intel i7 processor; with the process being fully automated once the files have been selected, this can be performed in the background while continuing other work. The VSA software was initially intended simply to parse all data files from a specified directory, outputting the raw velocity data in a format suitable for import into a spreadsheet. However, it soon became clear that a filtering algorithm was required to remove “spikes” of invalid data and that a simple graphical display of mean value data would be beneficial to allow quasi-real-time inspection of the data during the experiments. In its current form, the VSA software offers major advantages to the user via automating the parsing of the raw data files, filtering of invalid values, calculation of 3-D mean velocity distributions, calculation of turbulence characteristics such as turbulence intensity and Reynolds stresses, and determination of data-point and data-set mean values. The software also provides graphical display of the results, with parameter values automatically plotted at the appropriate position of the cross-section. To facilitate additional processing using software such as Matlab and spreadsheets the filtered time-series may be exported to either Matlab (.mat) or plain text (.dat) files.

Although the software was originally written to parse data files produced by the Nortek PolySync software, it is easily extendible to other formats and can also import data from other Nortek probes (such as Vector, Vectrino and Vectrino II), Turbulent Flow Instrumentation “Cobra” probes and data held in delimiter-separated (e.g. comma- or tab-separated) plain text files. Similarly, it now supports a wide range of filtering algorithms, from simple limiting values based on the time-series standard deviation through to Phase-Space Thresholding and Velocity Correlation methods (these are described briefly later), and includes additional display functionality and more advanced processing such as quadrant-hole analysis and power spectra calculation. Recent modifications include allowing the definition of custom, non-rectangular cross-sections, the addition of trim functionality for the data-point time-series, and the automated splitting of large data files into a number of shorter time-series. The VSA is downloadable as free software from www.mikejesson.com.

Following this introduction are two sections which describe the VSA. The first introduces the functionality which it provides, whilst the second discusses the implementation of this functionality. This latter section will be of interest to users as it describes how the raw data files are handled and the files created by the VSA.

2. The user perspective

In order to give an overview of the functionality of the VSA, this section describes the steps involved in creating a data-set, importing data from the raw data files, and examining the calculated parameters. It is necessarily brief and does not cover all functionality – for more details the reader is referred to the user guide [9]. Hereon, “probe” signifies a part of an instrument which is measuring velocities at a single point.

2.1. Configuration

Before importing data a number of data-set parameters must be set. These include characteristics of the cross-section (e.g. left- and right-bank position for rectangular channels, or the specification of a boundary definition file for more complex cross-sectional shapes) and filtering options. If multiple probes are being used then the relationship between the probe positions should be specified. In multi-probe scenarios, the “main” probe traverses the data-point positions, while the second probe may be either “fixed” (i.e. held at a constant absolute position) or offset (i.e. maintains a constant position relative to the main probe). The former case may be used if, for example, a single position is used as a baseline to verify that the flow does not change during the course of the experiment, while the latter allows faster traversing of a measured section by measuring multiple data-points simultaneously. More than two probes may be used, allowing for multiple offset probes, though only one probe can be marked as fixed. Default values for these parameters are set via the configuration dialog (Fig. 2); any data-set created will take those values, although the settings for individual data-sets are customisable through their own configuration screen.

2.2. Data-set creation

Four types of data-set may be created. The simplest is a single-probe data-set, in which each raw data file contains data from one probe only. In the case of the PolySync software, for example, data from multiple probes is recorded to a single file. For such systems, a multi-probe data-set should be created. When importing multi-probe data, the position of each probe within the cross-section is determined from the relationship specified in the configuration dialog.

In the authors’ current work, simulating thunderstorm downbursts, a multi-run approach is required. The velocity time-series associated with a downburst is highly unsteady and both full-scale and laboratory data show significant variation between downburst events or experimental runs, although the large-scale features are consistent (see [10–12]). It is therefore appropriate to represent a “typical” time-series by the ensemble-mean of time-series gathered over a number of experimental runs, here termed the “run-mean”. The VSA incorporates multi-run functionality, producing a time-series which is the run-mean of a number of synchronised time-series. The individual run time-series are synchronised by offsetting each run such that the measurement which is the first exceedence of a limiting value (or the time of the maximum value if preferred) occurs at the same time for each run. To use this functionality, a multi-run data-set should be created; as for single-run data-sets, multi-run data-sets may be single or multiple probe.

2.3. Importing from raw data files

There are three overarching file types which may be imported:

1. Delimiter-separated files of u , v and w measurements (these being the streamwise, lateral and bed-normal velocity components respectively).

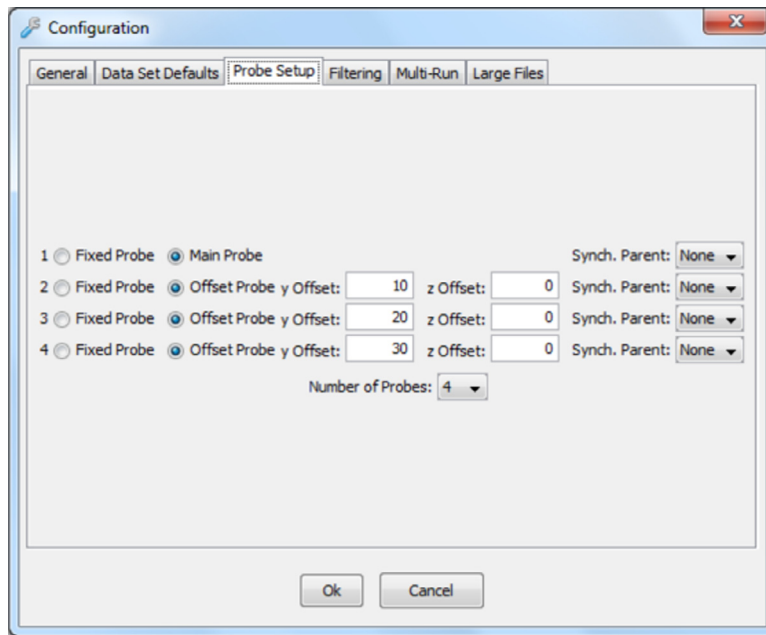


Fig. 2. Configuration dialog.

2. “Converted” files – Nortek, for example, provides software which will read their binary format and output the data as ASCII files.
3. Binary format.

For the purposes of streamlining the data processing procedure, import of the binary files is preferred as all processing can then be performed by the VSA, rather than having to use proprietary software to convert the files. A full list of the supported file types may be found in the user guide [9] but includes:

- Nortek PolySync, Vectrino, Vector and Vectrino-II files.
- Converted (NDV) versions of the Nortek binary files.
- Turbulent Flow Instrumentation “Cobra” probe files.

Other file formats may be supported with minor development of the software, as discussed later in Section 3.

Once the data-set has been created as described in Section 2.2, the relevant import buttons appear, corresponding to the file types described above. Clicking one of these buttons will bring up the import dialog (Fig. 3). Data files in the selected directory are shown if they have a valid file extension (e.g. .vno for Nortek binary files) and if their filename is of the correct format. The basic filename format is $\langle y\text{-coord} \rangle\text{-}\langle z\text{-coord} \rangle.\langle extension \rangle$ and is important as it is used to position the data within the cross-section, using the coordinates specified in the filename.

To start the import, files may be selected and then “Import Selected” pressed, or pressing “Import All” will import all files within the directory. During the import process, the data is filtered using the specified despiking method (see Section 2.4). Once files have been imported, the mean streamwise velocity at each point is displayed graphically (Fig. 4); the other components may be seen by switching to the relevant tab on the display. The data is displayed in tabular form below the graphical display, in the data-point summary table.

2.4. Despiking

The removal of invalid data from velocity time-series is essential if turbulence parameters are to be calculated accurately. “Despiking” is used here to refer to the combination of a filtering algorithm, which identifies invalid data “spikes”, and a spike replacement method, which replaces the removed invalid data with statistically valid values. A discussion of the sources of these invalid data

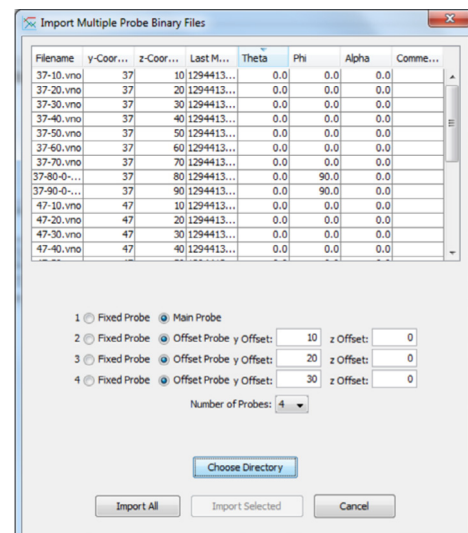


Fig. 3. Import dialog.

is outside the scope of this paper, and will depend on the instrument used to gather the data. The references provided below for the filtering algorithms all contain good discussions of the sources of spikes.

The VSA provides the following filtering options:

- Exclude level – values outside of a user-specified multiple of the standard deviation are identified as spikes.
- Velocity correlation (VC) [13] – the velocity fluctuations in each dimension are plotted against one another and a bounding ellipse, formed by the statistical limit of valid values, is constructed. Values which fall outside the ellipse are identified as spikes.
- Phase-space thresholding (PST) [7] – a bounding ellipse is used, in a similar manner to VC, but the velocity fluctuations and their first and second derivatives are plotted, rather than the fluctuations in the three dimensions.
- Modified PST [14] – as for PST but an upper limit for valid values is set which is used to identify large magnitude spikes prior to the application of the PST method, and a second limit is set which identifies values which must never be identified as spikes

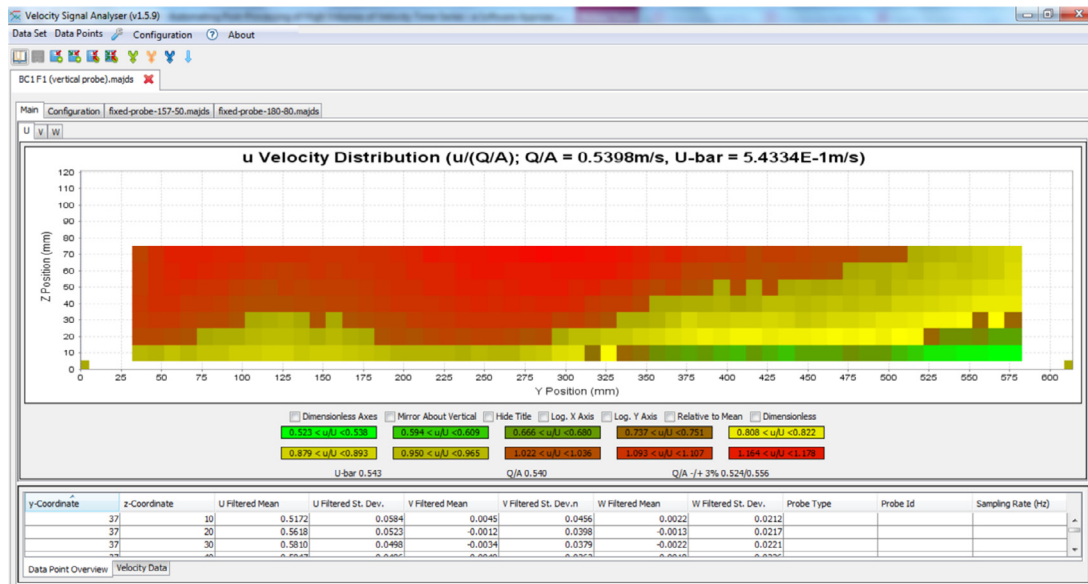


Fig. 4. The main window after import.

during the PST process. The aim of these is to reduce instances of valid values adjacent to spikes being identified as spikes due to the large magnitude of the first and second derivatives.

- Correlation and signal-to-noise ratio (SNR) – where the signal correlation and SNR are recorded by the instrument these are compared to user-specified limiting values. If either is below their respective limit then the associated value is identified as a spike.
- 50 point moving average – this is not technically a filtering method, but smooths the time-series.
- “w1 and w2 Difference” – where the instrument provides two independent measurements (w1 and w2) along one axis (e.g. Nortek Vectrino), the user may specify a limiting value, relative to the standard deviation of the w1 time-series. If the absolute difference $|w2 - w1|$ exceeds this then the values for all velocity components are identified as a spike.

To replace the spikes, a choice of spike replacement methods is provided:

- Linear interpolation between the valid values either side of the spike.
- Last good value (sample-hold) – the last valid value is used to replace the spike.
- 12-point Polynomial Interpolation – the best fit polynomial through the valid values surrounding the spike is calculated, and the replacement value interpolated [7].

2.5. Data-point details

Double clicking on a row in the data-point summary table will bring up the details for that data-point, including filtered and unfiltered time-series for all velocity components, tabular display of the time-series values and statistical properties (Fig. 5).

2.6. Data-point manipulation functionality

Additional data-point functionality is available by selecting rows in the data-point summary table and right-clicking. The menu shown gives access to a number of tools, including:

- Data-point selection tool, simplifying the selection of large numbers of data-points.
- Rotation correction, which corrects for errors in probe alignment (please see the user guide [9] for an explanation of this functionality).

- Save details to file, allowing the filtered time-series to be output as a plain text file, or in .mat format for import to Matlab.

2.7. Time-series analysis

The VSA software also gives access to analysis tools for the individual data-point time-series, including power spectra, Quadrant-Hole analysis and probability density functions. First introduced by Lu and Willmarth [15], the aim of quadrant-hole analysis is to identify turbulence propagating events and categorise the velocity measurements into different types of “event” (e.g. ejections and sweeps), allowing the contribution to Reynolds stress of each event type to be determined. To categorise the time-series of velocity measurements, simultaneously measured velocity fluctuations are used as the coordinates, $(u'(t), w'(t))$, in the u' - w' plane. The measurements are thus split into four event types, each corresponding to a quadrant of the u' - w' plane. Events in quadrant i are termed quadrant i events. Propagation of turbulence is indicated by a predominance of ‘ejection’ events (a quadrant 2 event in which the fluid is moving relatively slowly in the streamwise direction ($u' < 0$) whilst also moving upwards, $w' > 0$) and ‘sweep’ events (a quadrant 4 event in which $u' > 0$ and $w' < 0$). A full description can be found at [2].

To allow comparison of the flow over the cross-section, multiple data-points may be selected and their data are presented together as shown in Fig. 6.

2.8. Parameter inspection

In addition to the basic velocity data and the analyses described above, the VSA calculates:

- Lateral velocity vectors
- Turbulence intensity
- Turbulent kinetic energy and its flux
- Vertical and horizontal Reynolds stresses
- Momentum transfer

These data are presented in graphical form, and are accessed through a right-click menu from the velocity graph on the main screen. This menu also gives the option of displaying information regarding the quality of the data, by showing the percentage of measurements in each time-series which were considered valid and not removed by the chosen filtering algorithm. The graphs present display options, such as logarithmic or dimensionless axes and scaling

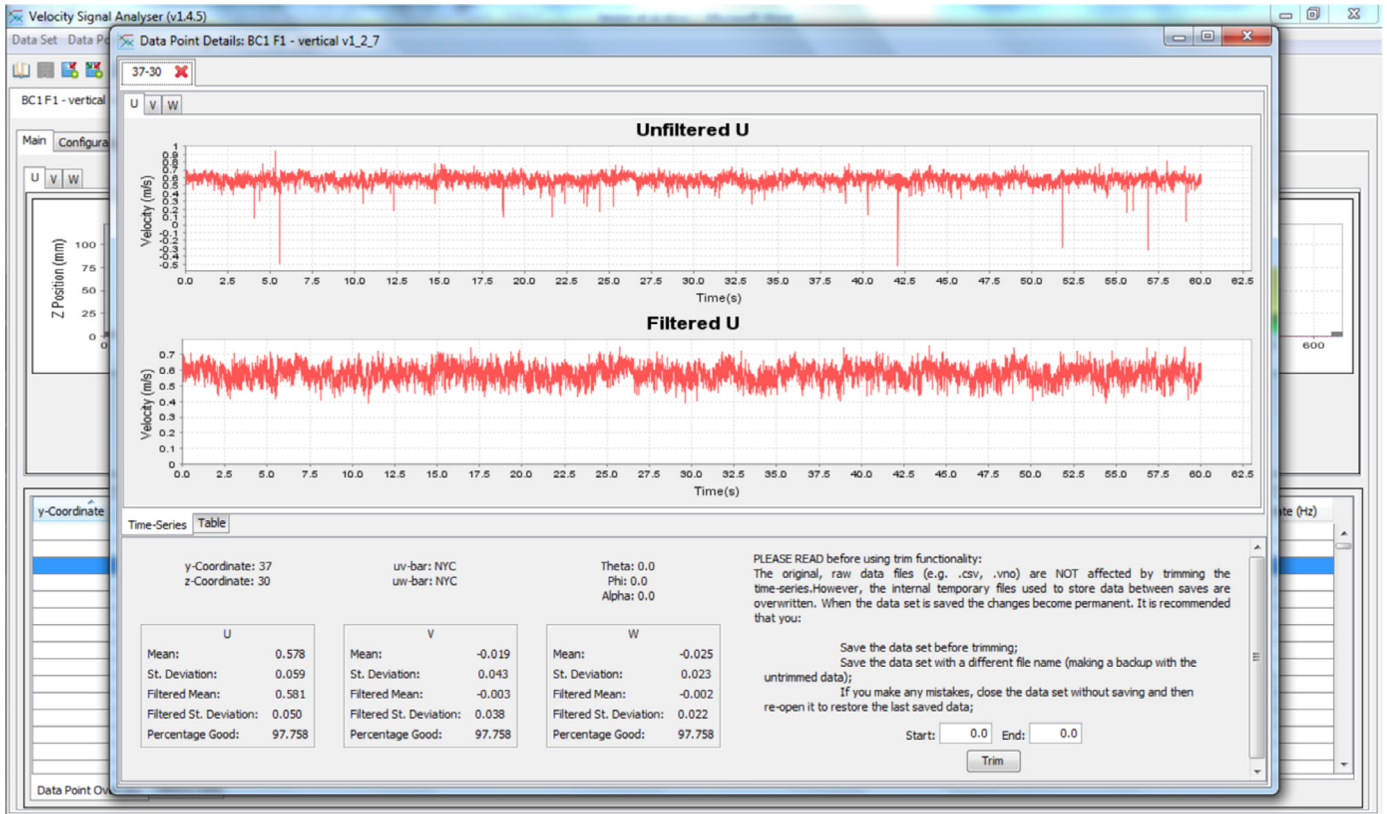


Fig. 5. Data-point details display.

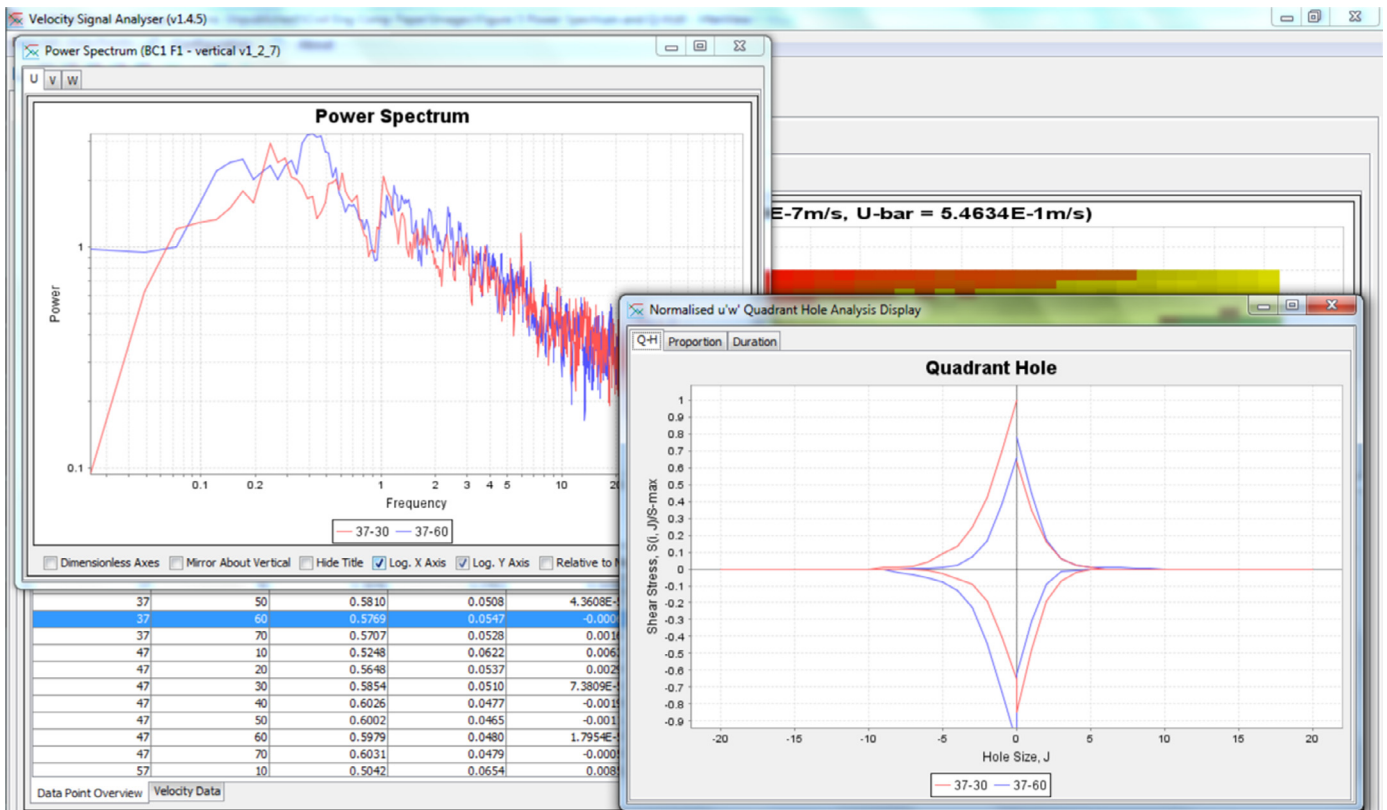


Fig. 6. Power spectra and quadrant-hole display.

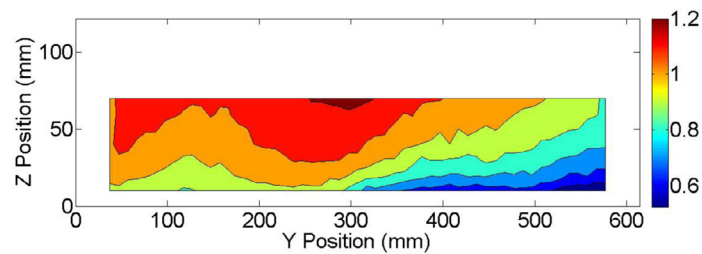


Fig. 7. Example matlab contour plot of u velocity distribution.

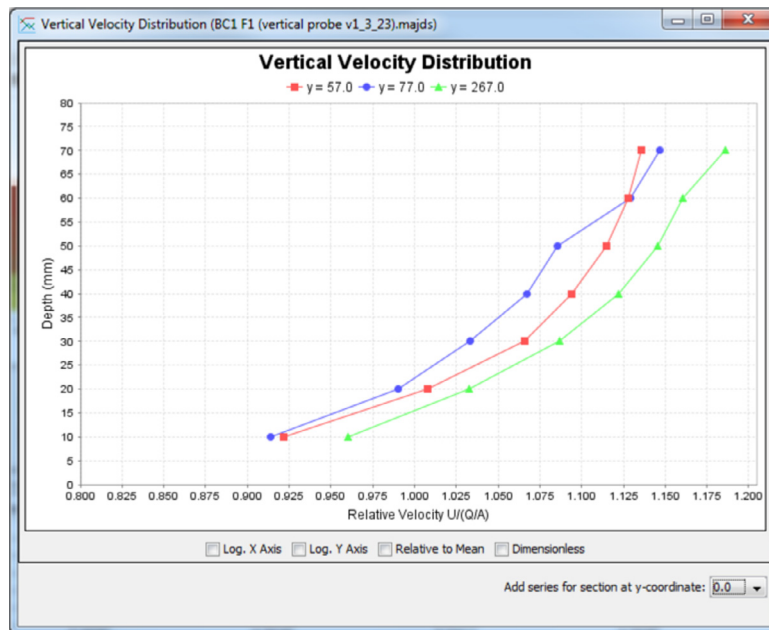


Fig. 8. Vertical section graph.

by the data-set mean of the parameter (in the case of the velocity graphs, the scaling uses the mean streamwise velocity calculated from the cross-sectional area by continuity of mass).

All graph data can be exported, either as a table from which data may be copied and pasted into a spreadsheet, or as a Matlab script which may either be saved or copied into Matlab. For the graph shown in Fig. 4 the Matlab script produces Fig. 7.

The VSA also allows the spawning of vertical (or horizontal) section graphs shown in Fig. 8, from the contour plots. These are Cartesian graphs of parameter value against vertical position, with distributions shown for user selected horizontal position. These allow easy comparison of vertical distributions at locations across the measured cross-section. As with the contour plots, these may be exported in tabular form and copied into a spreadsheet.

Whilst the applications shown here relate to water engineering, the VSA has also been successfully applied to the study of time series data arising from wind engineering research, and thus has extensive potential applications in the environmental engineering field.

3. Software design

3.1. Overview

The VSA is written using the Java programming language. Java was chosen partly due to the lead author's experience in graphical user interface (GUI) development using Java but also to allow the creation

of a platform-independent software package. The source code is organised into the “frontEnd” (the GUI) and the “backEnd” (the data processing code), which communicate using an application programming interface (API). It is therefore feasible to produce a bespoke GUI, with the data-processing performed by the existing code.

Internally, the data structure is independent of the format of the raw data file. Adding additional file types is simply a matter of implementing an import function for the new format, which will parse the raw data file and convert it into the internal format. It is therefore easy to extend the VSA to support other file formats.

Once read from the raw data files, the VSA stores the data in its own files. Thus, for security of experimental data, the raw data files are never written to by the VSA. Due to the large amount of data created in experimental work using instruments which record at a high frequency, the individual time-series are only held in memory for as long as they are required. As each data file is read, summary data (for example, the mean velocity) is calculated and this is stored in a data-set summary file (with a “.majds” extension). For each data-point, the time-series data are stored in a data-point details file and are only read into memory when the time-series are needed for display, calculation or export. The data-point files may be stored as xml format (with a “.majdp” extension), or in a binary format (with a “.majdpb” extension). The binary format is a recent addition, motivated by the very large quantities of data created by the authors' current work, in which a single data-set has a size of approximately 80 GB – this reduces by a factor of 10 when the binary format is used, thus making the VSA an appropriate and efficient data management tool.

3.2. External libraries

While the majority of the VSA code has been written specifically for the VSA, a number of external libraries are used to provide functionality. The graphs are based around the JFreeChart library [16], with additional functionality created by the authors. Export to Matlab files and saving in binary format are provided via a modified version of the JMatIO library [17], with the modifications by Robert Craig at Nortek and the authors. Some mathematical functionality, in particular the polynomial fitting used in spike replacement algorithms, is provided by the JAMA matrix library [18].

3.3. Discussion

The VSA software provides a streamlined mechanism for processing and inspecting velocity time-series data. While the majority of the functionality could be implemented using spreadsheets or software such as Matlab, the VSA provides an intuitive, GUI-driven solution to the problem of converting velocity data into useful information. In place of, for example, a large number of inter-linked spreadsheets, the VSA provides a single point of access to the data. Changes to the processing scheme, such as a change of filtering method, can be applied to all the data-points in a data-set with a few mouse clicks, and the newly processed data saved as a separate data-set for comparison. When a large number of data-points are gathered, the VSA allows the user to quickly and easily compare data from a number of different points or vertical/horizontal sections, rather than having to manually extract the data from different files.

Although the VSA was originally written to process data gathered in a rectangular flume, the ability to specify custom cross-sectional profiles makes it suitable for both field and laboratory measurements. In their experience, the authors have found the automated import and graphical display of velocity data to be invaluable, giving the capability for quasi-real-time inspection of data. This means that data can be parsed while the next measurements are being made, allowing the user to note areas of interest (or issues with the experimental procedure) during the experimental process, rather than at a later date when repeatability may be an issue. Alternatively, by allowing the batch-processing of all (or selected) data files within a directory, the import process may be started and allowed to run in the background if a large number of data files are to be imported together.

While a very small number of the calculations reflect the VSA's origins as a tool for data gathered in a hydraulics laboratory (for example, the fluid density used for boundary and Reynolds stress calculations is that of water; given sufficient interest, it would be a simple task to make this user-definable), the vast majority of the functionality is fluid-independent. The multi-run synchronisation functionality, for example, was implemented for the authors' current work on thunderstorm downbursts, and the filtering methods are based either on signal quality data provided by the instrument or statistical methods and are thus equally applicable to fluids other than water.

In addition to allowing inspection of the data, the VSA has been designed to facilitate the production of publication-quality figures via Matlab or a spreadsheet. Along with the ease with which comparisons may be made across the data-set, this functionality is expected to be of benefit to both academic researchers publishing journal papers and industrial users producing reports.

4. Conclusions

The VSA software when developed in response to the need to automate the parsing of data files of velocity time-series, and the post-processing of the data. The VSA has proven to be particularly useful when large numbers of data files are to be processed.

Using the VSA, a large number of parameters, such as mean, 3-D velocities, Reynolds stresses, turbulence intensity, etc., are automatically calculated and data can be filtered using one of a number of built-in spike identification algorithms. Filtering techniques include exclusion, velocity correlation, phase-space thresholding (PST), modified PST, Correlation and signal-to-noise ratio, 50 point moving average, w1 and w2 difference. De-spiking techniques include linear interpolation, last good value, and 12 point polynomial interpolation.

Whilst the applications shown here relate to water engineering, the VSA has also been successfully applied to the study of time series data arising from wind engineering research, and thus has extensive potential applications in the environmental engineering field. The VSA has greatly aided water and wind engineering research at the University of Birmingham, and is now in use by a number of researchers at other institutions.

Acknowledgements

The authors would like to thank Robert Craig at Nortek for supplying a modified version of the jmatio java library, along with the original developers of jmatio. Thanks must also go to the authors of the JFreeChart and JAMA libraries, which have proved invaluable in the development of this software. The original research was funded by an Engineering and Physical Sciences Doctoral Training Grant.

References

- Jesson MA, Bridgeman J, Sterling M. Automating post-processing of high volumes of velocity time-series: a software approach. In: Topping BHV, Iványi P, editors. Proceedings of the fourteenth international conference on civil, structural and environmental engineering Computing, Civil-Comp Press, Stirlingshire, UK; 2013. Paper 234.
- Jesson MA. The effect of heterogeneous roughness on conveyance capacity and application to the shiono-knight method, UK: University of Birmingham; 2012. [PhD Thesis].
- Jesson MA, Sterling M, Bridgeman J. An experimental study of turbulence in a heterogeneous channel. *Proc Inst Civil Eng Water Manag* 2013;166(1):16–26.
- Jesson MA, Sterling M, Bridgeman J. Modelling flow in an open-channel with heterogeneous bed roughness. *J Hydraulic Eng* 2013;139(2):195–204.
- Shiono K, Knight DW. Two dimensional analytical solution for a compound channel. In: Iwasa Y, Tamai N, Wada A, editors. Proceedings of the 3rd international symposium on refined flow modelling and turbulence measurements. Tokyo: Universal Academy Press; 1988. p. 503–10.
- Shiono K, Knight DW. Turbulent open-channel flows with variable depth across the channel. *J Fluid Mech* 1991;222:617–46.
- Goring DG, Nikora VI. Despiking acoustic doppler velocimeter data. *J Hydraulic Eng* 2002;128(1):117–26.
- Jesson MA, Sterling M, Bridgeman J. Despiking velocity time-series – optimisation through the combination of spike detection and replacement methods. *Flow Meas Instrum* 2013;30(April):45–51.
- Jesson MA. Velocity Signal Analyser – Installation and User Guide; 2012.
- Fujita TT. Tornadoes and downbursts in the context of generalized planetary scales. *J Atmos Sci* 1981;38(8):1511–34.
- Chay MT, Letchford CW. Pressure distributions on a cube in a simulated thunderstorm downburst – Part B: moving downburst observations. *J Wind Eng Ind Aerodyn* 2002;90(7):733–53.
- McConville AC, Sterling M, Baker CJ. The physical simulation of thunderstorm downbursts using an impinging jet. *Wind Struct* 2009;12(2):133–49.
- Cea L, Puertas J, Pena L. Velocity measurements on highly turbulent free surface flow using ADV. *Exp Fluids* 2007;42(3):333–48.
- Parsheh M, Sotiropoulos F, Porte-Agel F. Estimation of power spectra of acoustic-doppler velocimetry data contaminated with intermittent spikes. *J Hydraulic Eng* 2010;136(6):368–78.
- Lu SS, Willmarth WW. Measurements of the structure of the reynolds stress in a turbulent boundary layer. *J Fluid Mech* 1973;60(3):481–511.
- Object Refinery Limited, JFreeChart, [online]. Available from <http://www.jfree.org/jfreechart/>, 2012, [Accessed: 28/12/2012].
- The MathWorks Inc., JMatIO - MatLab's Mat-file I/O in JAVA, [online]. Available from <http://www.mathworks.co.uk/matlabcentral/fileexchange/10759>, 2007, [Accessed: 28/12/2012].
- The MathWorks Inc / National Institute of Standards and Technology (NIST), JAMA: A Java Matrix Package, [online]. Available from <http://math.nist.gov/javanumerics/jama/>, 2012, [Accessed: 28/12/2012].