13th Australasian Fluid Mechanics Conference
Monash University, Melbourne, Australia
13-18 December 1998

HIGH VOLUME LDA EXPERIMENTAL DATA ANALYSIS USING AN OBJECT-ORIENTED APPROACH

Johannes P. VAN DER WALT, Yos S. MORSI and Wei YANG

School of Engineering and Science
Swinburne University of Technology, Hawthorn, Victoria, AUSTRALIA

ABSTRACT

The processing and analysis of large volumes of quantitative data that can be gathered by fluid flow diagnostics systems such as Laser-Doppler Anemometry systems and other computerised data acquisition systems can pose quite a challenge. Often, it is required to establish time-mean and RMS values of several cyclic parameters that have been measured continuously with time. Writing a simple routine in a procedural language can be more difficult than first imagined. The paper demonstrates an object-oriented approach that is surprisingly simple, yet extremely flexible and can handle many megabytes of raw data in a flash.

INTRODUCTION

Experimental analysis of complex processes is a very important and widespread practice in engineering. It is often not practical, if not impossible, to accurately model the problem at hand analytically or numerically. One example of this is artificial heart valves where research and development relies heavily on experimental analysis to determine the characteristics and performance of different types of valves and varying material qualities. Whilst a large part of the activity is centred around the gathering of experimental data using a computerised Laser-Doppler Anemometry (LDA) acquisition system, enormous quantities of raw data can easily be generated.

In the case of the artificial heart valve, the hydrodynamic performance of the valve is determined by measuring the instantaneous blood velocity in three components. These measurements are recorded at specific points around the artificial valve to establish shear stresses, accelerations and decelerations which may lead to the damaging of blood cells and the formation of haemolysis. Changes in the heartbeat rate, and hence changes in the deformation rate of the artificial valve materials, may also cause material properties to change for some valve materials. To analyse this behaviour, tests with accelerations and decelerations of the heartbeat rate have to be done. This implies that the time steps at which LDA readings are taken will have to be varied substantially. Furthermore, decay in the valve material properties with time can alter the behaviour of the valve. Typically, velocities are measured in time steps of one one-hundredths of a second for periods of several hours, and in the case of material decay testing, several weeks in accelerated tests with even shorter time steps.

Since heartbeats are a cyclic phenomenon, the data gathered with time is in essence a composition of numerous sequential tests with data gathering being repeated many times at the same (or almost the same) valve position in the cycle. Typical results are shown in Figure 1. Note that the values at 0.1 and 0.5 seconds represent the same cyclic position. Similarly, the values at 0.35 and 0.85 seconds also represent the same cyclic position. Values at the same cyclic position have to be averaged in the case of steady state testing, but material property changes with time may also be observed at the same cyclic position. There are numerous statistical packages available that could be used for this purpose such as SAS[1], SPSS[2] and even Access[3], however, some of them are extremely expensive and generally engineers and scientists are not familiar with them. These packages have the further disadvantage that the user has to have continued access to it to be able to do data analysis. An increasing number of engineers and scientists are becoming familiar with object-oriented languages such as C++, Delphi and, even though Visual Basic is not a true object-oriented language, it too may be used.

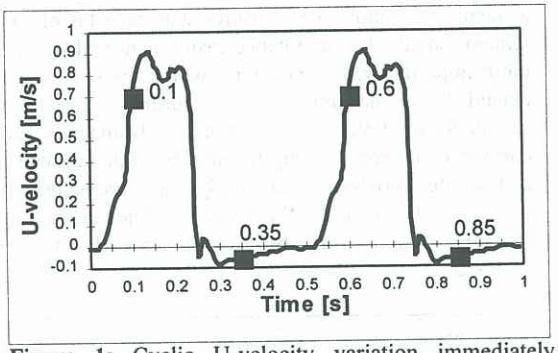

Figure 1: Cyclic U-velocity variation immediately downstream of an artificial heart valve measured with a computerised LDA system.

The aim of this paper is to develop an object-oriented routine for educational purposes. The routine will be able to read large files of raw experimental data generated by the LDA system, calculate time-mean and RMS velocity components for a certain test period for which results are to be averaged, calculate the shear stresses at each time step of the cycle and repeat the process for the next set of input data. Ultimately, the time-mean and RMS values, shear stresses and other calculated parameters are written to a disk file for importation into a plotting package.

OBJECT-ORIENTED DESIGN OF THE ROUTINE

The Bin Object

The object-oriented design is centred on the concept of several bins (containers), each used to store results within a narrow specified time step band. This concept is

shown by Figure 2. The raw experimental data retrieved from the hard disk is passed through a timeslot-sorting algorithm that will pass the raw data on to the appropriate bin. Since each bin is an object, data storage, manipulation and output is handled by the object. The benefit of this is that the complexities of these actions are not visible to the rest of the program. Each bin also contains the final time-mean and RMS values and calculated parameters required such as shear stress for that specific time slot which really translates to a specific position of the valve movement cycle.

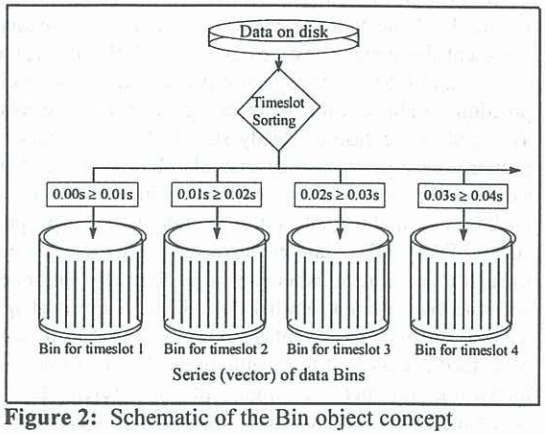

Figure 2: Schematic of the Bin object concept

As will be shown later, the Bin objects are contained in another object of which the timeslot-sorting algorithm is a member function. This ensures that each bit of logic remains small, clear and hence easily manageable. Even more importantly, the ease with which the code can be extended to accommodate a further variable or calculations holds tremendous advantages over conventional programming techniques. For instance, to add another variable would simply imply the addition of this variable to the Bin object. The design and functionality of the bin object is schematically shown in Figure 3.

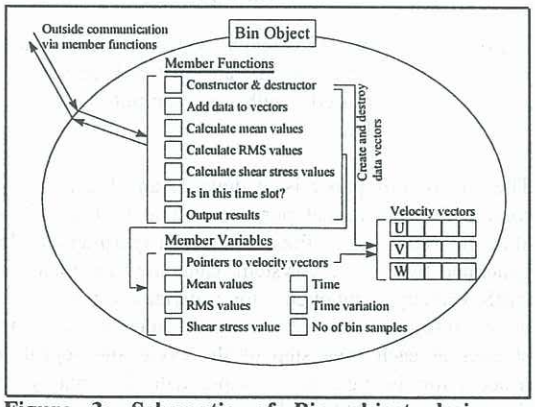

Figure 3: Schematic of Bin object design and functionality

The Bag Object

In the *Bin Object* section the concept of several bin objects, each containing the data for a specific time slot, was discussed. To manage and correspond with these bin objects, a bag object is created. The bag object contains all the bin objects and several functions and variables required to properly manage the bin objects. Furthermore, the bag object provides for user interaction and data file input and output. Figure 4 provides a schematic representation of this object:

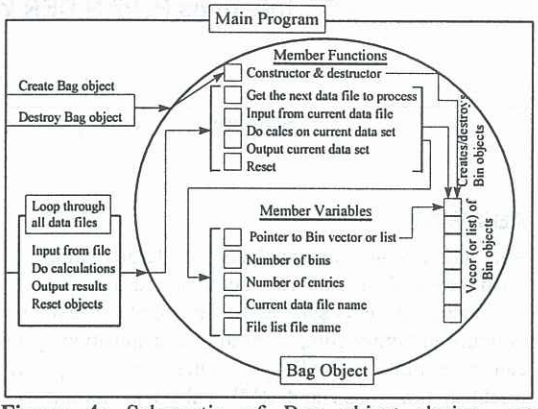

Figure 4: Schematic of Bag object design, user interaction and Bin object manipulation.

IMPLEMENTATION

The program was coded in C++ using Microsoft Visual C++ 4.2 and compiled as a *Console Application* under Windows 95 for simplicity. Total code length is 185 lines and took four hours to design and implement from scratch.

PROGRAM PERFORMANCE

Data files containing 20,000 entries each have been processed within 15 seconds on a 100 MHz Pentium PC with 32 MB of RAM.

CONCLUSION

Object-oriented techniques are ideally suited when vast quantities of complex experimental data have to be processed and analysed. The benefits over procedural codes are tremendous; especially in terms of the simplicity and flexibility of the code. Since an increasing number of engineers and scientists are becoming familiar with object-oriented programming techniques, this knowledge can be readily applied to solve some of their data analysis problems. This paper demonstrated the simplicity of the general code design and also showed that the object-oriented approach results in an extremely flexible routine that can be easily adapted and extended to fulfil other requirements. The C++ source code is available from the authors as freeware.

REFERENCES

1. SAS User's Guide: Basics, *SAS Institute Inc.*, 1990.
2. SPSS Reference Guide, *SPSS Inc.*, Chicago, 1990.
3. Microsoft Access for Windows 95, *Microsoft Corporation*, Redmond, WA, 1997.
4. BUZZI-FERRARIS, G., "Scientific C++, Building Numerical Libraries the Object-Oriented Way", *Addison-Wesley*, 1993.
5. BARTON, J.J. and NACKMAN, L.R., "Scientific and Engineering C++", *Addison-Wesley*, 1994.