Neural network reconstruction of fluid flows from tracer-particle displacements

G. Labonté

Abstract We demonstrate some of the advantages of using artificial neural networks for the post-processing of particle-tracking velocimetry (PTV) data. This study is concerned with the data obtained after particle images have been matched and the obvious outliers have been removed. We show that it is easy to produce simple back-propagation neural networks that can filter the remaining random noise and interpolate between the measurements. They do so by performing a particular form of non-linear global regression that allows them to reconstruct the fluid flow for the entire field covered by the photographs. This is obtained by training these neural networks to learn the fluid dynamics function \( f \) that maps the position \( \mathbf{x} \) of a fluid particle at time \( t \) to its position \( \mathbf{X} \) at time \( t + \Delta t \). They can do so with a high degree of precision when provided with pairs of matching particle positions (\( \mathbf{x}, \mathbf{X} \)) from only about 2 to 4 pairs of PTV photographs as exemplars. We show that whether they are trained on exact or on noisy data, they learn to interpolate with such a precision that their output is within one pixel of the theoretical output. We demonstrate their accuracy by using them to draw whole streamlines or flow profiles, by iteration from a single starting point.

1 Introduction

Particle-tracking velocimetry (PTV) is a technique used in particle-image velocimetry (PIV) for determining the velocity field of a fluid in motion [see Adrian (1991)]. It consists in measuring the displacements undergone by small particles suspended in this fluid during a small time interval \( \Delta t \). In its simpler form, this technique is mainly used for two-dimensional fluid flows. Two photographs of the same region of the fluid are taken, from a direction at \( 90^\circ \) to its plane of motion, one at time \( t \) and the other one at time \( t + \Delta t \). These are then usually digitized so that a computer can automatically process them. It is necessary to find the position of the geometric center of the particle images, in some reference frame. There are many efficient procedures for doing so, e.g., the neural network method described by Carosone et al. (1995). Once this is done, one must solve the correspondence problem, which consists in determining which points, in the two photographs, represent the same particles. Labonté (1999) mentions some of the best methods used for this and describes a selforganizing map neural network that can very efficiently solve this problem. The present work deals with the processing of the data obtained once image matching is completed.

1.1 The inaccuracies in the data

As discussed, e.g., in Adrian (1991), Westerweel (1994) and Luff et al. (1999), the errors in the measurements of PTV and, more generally of PIV, are essentially of two kinds. There are errors that result from false matches between particle pairs in PTV and, with correlation methods in PIV, from the selections of false correlation peaks. These errors yield spurious vectors in the velocity field. The tip of such vectors is at a random, uniformly distributed, position in the interrogation window (of PTV or PIV). Many of them are thus much larger than their neighbors or have very different directions from them so that they can easily be recognized as outliers. Figure 1 illustrates this point; it shows displacement vectors as would be obtained in PTV, after matching two photographs of 200 particles each, each one having 20 particles that are unmatchable due to out of plane motion. The flow is that in a vortex with complex potential \( V(z) = iz \) that is singular at the origin. The number of spurious vectors present evidently depends on the efficiency of the matching algorithm. For the singular flow shown in Fig. 1, which is a difficult matching problem, there were 24 outliers out of 180 vectors, i.e., about 13% of outliers. Westerweel (1994) mentions the figure of 5% as typical for PIV data, and Sun et al. (1996) gives 15% for a difficult problem with high speed flow in a combustion chamber.

The other kind of errors in PIV result from the pre-processing of the pictures, when the background is removed and the intensity of the particle images is enhanced, from the finiteness of the grain of the film or of the pixel size in light detector arrays, and from the noise in the electronic device that digitizes the data. With correlation methods, there are also inaccuracies due to the deformation of particle-image patterns. A small number of

Received: 23 November 1998/Accepted: 14 July 2000

G. Labonté
Department of Mathematics and Computer Science
Royal Military College of Canada, Kingston
Ontario K7K 7B4, Canada
e-mail: labonte-g@rmc.ca
Fax: +1-613-3845792

This research was financed in part by a grant from the Academic Research Program of the Department of National Defense of Canada.
errors also correspond to the spurious vectors that were left in, after the attempt at their removal, because they were very close to the actual vectors. All of these errors are much smaller than those corresponding to outliers and constitute essentially random noise. In PTV, they lead to measured positions of the centroids of the particles \( \{ X_i \} \), being off by some random error vector \( \varepsilon_i \), so that, even in a correctly matched image pair, the measured displacement \( D_i \) is actually \( D_i + \varepsilon_i \), where \( D_i \) is the true displacement. One can assume as in Westerweel (1994) that the components of \( \varepsilon_i \) are statistically orthogonal and are distributed normally about the value 0 with standard deviation \( \sigma_\varepsilon \). This author mentions that, for most PIV systems, \( \sigma_\varepsilon \) is estimated at less than 1% of the full displacement range, which is defined as \( \text{max} - \text{min} \) (length of the maximum displacement) – (length of the minimum one).

Huang et al. (1993) mentions that locating the center of intensity of particles in PIV is usually done with an error under 0.25 pixels. Luff et al. (1999) test their data processing methods with artificial data having random, uniformly distributed, 1%, 3.3% and 4.5% relative absolute error in the velocity vectors. For particle-image displacements of 10–20 pixels, the largest errors are then about 0.5–0.9 pixel. These figures are somewhat higher than those of other authors, as quoted in Guézennec and Kiritis (1990). Guézennec et al. (1994) mention that in their tests a sub-pixel accuracy of the average of 0.3 pixel is normal.

1.2 Spurious vector removal

A large proportion of the spurious vectors can easily be eliminated manually, as mentioned in Westerweel (1994) and Sun et al. (1996). Upon so doing with Fig. 1, we removed successfully 20 out of the 24 spurious vectors. The four that we did not remove happened to be, by accident, close enough to the correct displacement vectors that they could be mistaken for these. This similarity to actual displacement vectors implies that the continuity of the displacement field will not be much altered by their being left in. Efficient algorithms, based on the continuity of the displacement vector field, exist to remove the outliers; see for example Landreth and Adrian (1990), Willert and Gharib (1991), Guézennec et al. (1994), Westerweel (1994), Hartman et al. (1996), and Song et al. (1999). As in the case of manual outlier removal, these methods are efficient enough that the only missed vectors are close to the actual displacement vectors and the displacement field is not appreciably distorted by their being left in.

1.3 Random noise removal

Once outliers have been removed, there is often need for further processing. This may be done to improve the accuracy of physical variables that are to be calculated from the velocity field, such as the vorticity (see Lourenco et al. 1995; Sun et al. 1996; Luff et al. 1999). It can also be done for the important purpose of flow field visualization, in which case, post-processing aims at improving the appearance for human observers of the graphical representations of the velocity field. Algorithms are then needed for the smoothing of the data and its interpolation.

In most published work in PIV, smoothing of the data is done simply by convolving the velocity field with a Gaussian function (see, e.g., Luff et al. 1999; Sun et al. 1996; Hartman et al. 1996). This technique usually succeeds in filtering out some of the random noise. Luff et al. (1999) report results to that effect for data obtained by auto-correlation in PIV. In Hartman et al. (1996), the Gaussian filter is combined with an extended vector median filter, as described in Astola et al. (1990). The latter filter is used mainly to remove the outliers but it does at the same time cancel some of the random noise.

One should however be cautious with filters because there will actually be a loss, instead of a gain, of precision when they average or convolve data over too large a domain. Small-scale information is then destroyed by their blurring effect (see Luff et al. 1999; Astola et al. 1990).

1.4 Interpolation

Interpolation is used in post-processing mainly for filling up the holes left by outlier removal or where the matching algorithm found no matches. It is then usually done only locally for small patches of the displacement field. The interpolation techniques normally used are spline fitting and linear regression to least-square fit with some chosen functions. Typically, Sun et al. (1996) use quadratic functions to interpolate the missing velocity vectors and Luff et al. (1999) do local regression using a second-degree polynomial. For data visualization, they use spline fitting, as do most commercial software.

Attempts are rarely made at fitting the whole or a large region of the displacement field. Luff et al. (1999) show the result they obtained when doing a full field regression, with a fourth degree polynomial, for the vorticity field of an Oseen vortex. The poor quality of the fit obtained is evident; the maximum relative error is 226% and the surface average relative error is 55%.