2015-06-01

A physics-enabled flow restoration algorithm for sparse PIV and PTV measurements

Vlasenko, A

http://hdl.handle.net/10026.1/3810

10.1088/0957-0233/26/6/065301
Measurement Science and Technology
IOP Publishing

A physics-enabled flow restoration algorithm for sparse PIV and PTV measurements

Andrey Vlasenko Edward C. C. Steele

W. Alex M. Nimmo-Smith

Abstract

The gaps and noise present in Particle Image Velocimetry (PIV) and Particle Tracking Velocimetry (PTV) measurements affect the accuracy of the data collected. Existing algorithms developed for the restoration of such data are only applicable to experimental measurements collected under well-prepared laboratory conditions (i.e. where the pattern of the velocity flow field is known), and the distribution, size and type of gaps and noise may be controlled by the laboratory set-up. However, in many cases, such as PIV and PTV measurements of arbitrarily turbid coastal waters, the arrangement of such conditions is not possible. When the size of gaps or the level of noise in these experimental measurements become too large, their successful restora-

tion with existing algorithms becomes questionable. Here, we outline a new Physics-Enabled Flow Restoration Algorithm (PEFRA), specially designed for the restoration of such velocity data. Implemented as a “black box” algorithm, where no user-background in fluid dynamics is necessary, the physical structure of the flow in gappy or noisy data is able to be restored in accordance with its hydrodynamical basis. The use of this is not dependent on types of flow, types of gaps or noise in measurements. The algorithm will operate on any data time-series containing a sequence of velocity flow fields recorded by PIV or PTV. Tests with numerical flow fields established that this method is able to successfully restore corrupted PIV and PTV measurements with different levels of sparsity and noise. This assessment of the algorithm performance is extended with an example application to *in situ* submersible 3D-PTV measurements collected in the bottom boundary layer of the coastal ocean, where the naturally-occurring plankton and suspended sediments used as tracers causes an increase in the noise level that, without such denoising, will contaminate the measurements.

1 Introduction

Particle Image Velocimetry (PIV) and Particle Tracking Velocimetry (PTV) are two established methods for the measurement of instantaneous distributions of velocity components within an illuminated 2D sample area or 3D sample volume. In both cases, digital cameras are commonly used to record traces of particles suspended in the flow field. A pair of traces are yielded by two successive laser-sheet pulses or two successive camera frames in PIV and PTV, respectively. The displacements in all the particles (on an ensemble-averaged or an individual basis) are then divided by the fixed time delay between the two exposures, thus obtaining the corresponding velocity distributions.

While the idea of the PIV and PTV methods is simple, the noise and gaps present in experimental measurements typically affects the accuracy of the data collected (Westerweel, 1994, Raffel et al., 2007). The noise arises from errors connected with the characteristics of the particles and their representation in the images (Hart, 2000). A low seeding density complicates these issues, as well as any subsequent analysis (Cenedese and Querzoli, 1997, 2000, Stanislas et al., 2004).

In recent years, several methods have been developed for the denoising and restoration of such data; exploiting the statistical or the physical characteristics of the velocity flow field.

In statistical methods, individual vectors that depart from the ensemble of the recorded velocity flow field are identified and subsequently eliminated. Such data post-processing commonly consists of using global-mean, local-mean or local-median tests or using global histogram operators (Westerweel and Scarano, 2005, Raffel et al., 2007, Duncan et al., 2010). Here, it is assumed that locally-occurring errors are randomly scattered within the sample volume, and that a sufficient quantity of tracers are present for the outliers to be detected. These methods are used for their convenience, computational cost and ease of implementation. However, only individual vectors are eliminated and not the noise that exists homogeneously within the sample volume.

Concomitant issues relate to infilling gaps in experimental measurements, and are tackled after statistical denoising. The restoration of 'gappy' data commonly consists of using different types of interpolation, e.g. kriging, nearest neighbour or polynomial interpolation from linear to nth order (cf. Stuer and Blaser 2000). Similarly, methods that employ Proper Orthogonal Decomposition have gained popularity, remaining cost efficient while still being applicable to any type of flow (Venturi and Karniadakis, 2004, Gunes and Rist, 2008). These exhibit good restoration capabilities where the sparsity of these data are 50 %, but the performance decreases as the sparsity of the data approaches 20 %.

In physical methods, hydrodynamical equations, e.g. Navier-Stokes (NSE) or Vorticity Transport Equations (VTE), are used for the restoration of noisy *and* gappy data. Typically, this is achieved by fitting numerical pre-estimates of the (same) velocity flow field to data collected from experimental measurements using Kalman filtering (Suzuki, 2012) or variational methods (Okuno et al., 2000, Suzuki et al., 2009a,b), such that they are similar. Since the velocity data from these schemes are determined from the results of the numerical hydrodynamical model, the results of the restoration are physically-plausible yet are not limited by the occurrence of noise or the sparsity of the data. However, this is only feasible where numerical pre-estimates of the velocity flow field are possible (i.e. where boundary and initial conditions are known *a priori*).

Contrary to methods using numerical pre-estimates, Sciacchitano et al. (2012) suggested deriving boundary conditions directly from experimental measurements, that then are used to infill gappy data in a physically-plausible way. However, this is very sensitive to noise (Sciacchitano et al., 2012).

All these methods are able to be used for the denoising and restoration of experimental measurements within the context of a well-prepared laboratory set-up, where no unsuitable particles are present and tracers with known light scattering characteristics are selected and seeded in the velocity flow field.

Tuning laboratory settings (e.g. by optimising the concentration / size of the particles tracked) results in the permissible level of gaps and noise that allows successful restoration using existing methods. Even if gaps and noise cannot be sufficiently reduced, the laboratory set-up offers enough details that numerical pre-estimates are possible, as the boundary conditions or the pattern of the velocity flow field are known *a priori*. However, in several cases, it is not possible for these gaps and noise to be sufficiently reduced nor any pre-estimates to be made. An example of this is seen in PIV and PTV measurements in ocean flows (Nimmo-Smith et al., 2002, 2005, Nimmo-Smith, 2008) where the arrangement of usual experimental conditions using ideal tracers is not possible and naturally-occurring suspended particles are used instead. The uneven shape of these particles, scattered inhomogeneously within the velocity flow field, causes an increase in the occurrence of gaps and noise that, in turn, complicates any later analysis. In addition, as only the part of the ocean advected through the sample volume are recorded, the boundary conditions are unknown and numerical pre-estimates are not feasible. Therefore, restoration of such data with existing methods is debatable; requiring the development of a new Physics-Enabled Flow Restoration Algorithm (PEFRA) for these velocity measurements. This is founded on a hydrodynamical basis, as represented by the Vorticity Transport Equation (VTE), however it is independent of specified boundary conditions and the algorithm exhibits a weak sensitivity to noise, as confirmed by tests using both artificial/numerical and in-situ experimental data.

PEFRA is from the same pedigree as the Physically-Consistent and Efficient Variational Denoising (PCEVD) algorithm developed by Vlasenko and Schnorr (2010), but with a significant improvement that allows restoration of gappy and noisy data. Both methods conform to a black box philosophy, requiring no specific user-background in fluid dynamics (except in special cases) and may be applied to any velocity time-series, formed from any type of flow and corrupted by any type of noise. However, PCEVD is limited in the sparsity permitted, especially under turbulence. This failing is corrected in PEFRA, and confirmed by the restoration of a velocity flow field with only 10% of data available.

Here, PCEVD is outlined in §2, with the development of PCEVD into PEFRA outlined in §3. In §4, the algorithm sensitivity to noise and sparsity is discussed, with an assessment of the algorithm performance using artificial/numerical data modelling different flow conditions presented in §5. This assessment is extended to submersible 3D-PTV measurements in ocean flows, in §6, where naturally-occurring suspended particles are used as tracers. The pseudo-code outline of PEFRA is presented in Appendix B.

2 PCEVD algorithm

A detailed discussion of the mathematical background to PCEVD containing the complete proofs may be found in Vlasenko (2010) (or in compact form in Vlasenko and Schnorr 2010), and only a summary (without theoretical substantiation) is provided here as the context for the solution of the problem. To do so, $\vec{a}(\vec{x})$ and $\vec{b}(\vec{x})$ are defined as two vector functions in a volume, V, where $\vec{x} \in V$ is a three-dimensional coordinate vector. Then, assuming that $\vec{a}(\vec{x})$ and $\vec{b}(\vec{x})$ are differentiable, the L2 norm is defined as: $\|\vec{a}\|_2 = \sqrt{\int_V \vec{a}(\vec{x})^2 d\vec{x}}$, the inner product is defined as $\langle(\vec{a}, \vec{b})\rangle = \int_V (\vec{a} \cdot \vec{b}) d\vec{x}$ and the convolution of these is defined as: $\vec{a}(\vec{x}) \star \vec{b}(\vec{x}) = \int_{-\infty}^{+\infty} \vec{a}(\vec{x})\vec{b}(\vec{t} - \vec{x}) d\vec{t}$.

The curl, finally, is defined as: $\nabla \times \vec{a} = [\frac{\partial a_z}{\partial y} - \frac{\partial a_y}{\partial z}; \frac{\partial a_x}{\partial z} - \frac{\partial a_z}{\partial x}; \frac{\partial a_x}{\partial y} - \frac{\partial a_y}{\partial x}]$. Importantly, the VTE is yielded when this operator is applied to both the LHS and the RHS of the NSE:

$$\frac{\partial \vec{\omega}}{\partial t} + (\vec{\omega} \cdot \nabla)\vec{v} + (\vec{v}\nabla)\omega = \nu \triangle \vec{\omega} \tag{1}$$

where, $\omega = \nabla \times \vec{v}$, $\triangle = \nabla^2$ is the Laplace operator and ν is the viscosity.

The benefit in using the VTE over the NSE is that it does not contain pressure as an additional variable. For the sake of simplicity, the LHS of the VTE is denoted by an $\vec{e}$, i.e. $\vec{e}(\vec{v}) = \frac{\partial \vec{\omega}}{\partial t} + (\vec{\omega} \cdot \nabla)\vec{v} + (\vec{v}\nabla)\vec{\omega}$. This shorthand is especially useful when the VTE is presented in weak form, i.e.

$J(\vec{\omega}) = \nu\|\nabla \times \vec{\omega}\|_2^2 + 2\langle \vec{e}(\vec{v}_s), \vec{\omega}\rangle$. The weak form of the VTE reverts to the normal form of the VTE by differentiation by $\vec{\omega}$.

PCEVD is an iterative algorithm that was developed for the denoising and restoration of three-dimensional velocity time-series data recorded in PIV, PTV or other velocity measurements. This is implemented in four stages: Gaussian filtering, solenoidal projection (i.e. divergence removal, demanded by the continuity equation), vorticity restoration and velocity restoration. On each loop, the quality of this output is checked by a termination criteria. If this is not achieved, the process repeats using the results generated in the last output. The idea of this sequence is that high-frequency noise, as well as any divergence, is eliminated by Gaussian filtering and solenoidal projection, respectively. Any remaining noise is then eliminated by vorticity restoration, where the pattern of the vorticity flow field is also recovered (– if it is corrupted). Finally, the last part of the algorithm, velocity restoration, links the pattern of the vorticity flow field and the filtered pattern of the velocity flow field, providing an additional connection to the PIV or PTV data. These stages are detailed below, via the restoration of a gappy and noisy velocity flow field, v_m, recorded in an incompressible fluid.

2.1 Stage 1: Gaussian filtering

The restoration of the velocity flow field, $\vec{v}_m$, is initiated by Gaussian filtering:

$$\vec{v}_d = g \star \vec{v}_m, \qquad g = \frac{1}{(2\pi\sigma^2)^{3/2}} \exp\left(-\frac{\sigma^2}{2}|\vec{x}|^2\right) \tag{2}$$

where, $\vec{v}_m$ is the recorded velocity flow field, $\star$ is the convolution and σ is the variance governing the strength of the Gaussian filtering (discussed in Section 4) that removes high frequency noise. The filtered velocity flow field $\vec{v}_d$ is then passed to Stage 2 where the divergence is eliminated.

2.2 Stage 2: solenoidal projection

As it is assumed that this fluid is incompressible, divergence within the velocity flow field constitutes noise and must be eliminated. Therefore, $\vec{v}_d$ is the sum of the divergence (∇p) and the solenoidal ($\vec{v}_s$) velocity components, i.e. $\vec{v}_d = \nabla p + v_s$, to which the divergence operator may be applied giving:

$$\nabla \vec{v}_d = \triangle p \tag{3}$$

Solving Equation 3 with zero boundary conditions results in the divergence part, $\triangle p$. This is subtracted from $\vec{v}_d$, giving the divergence-free velocity flow field v_s (consistent with the continuity equation) passed to Stage 3.

2.3 Stage 3: vorticity restoration

The physical plausibility of the flow that was filtered in Stage 1 and Stage 2 is enforced by the VTE. This is done by minimising the functional:

$$J(\omega) = \|\vec{\omega} - \vec{\omega}_s\|_2^2 + \alpha\Big(\nu\|\nabla \times \vec{\omega}\|_2^2 + 2\big\langle \vec{e}(\vec{v}_s), \vec{\omega}\big\rangle_{\vec{\omega}}\Big) \tag{4}$$

where, $\vec{\omega}_s = \nabla \times \vec{v}_s$ is the vorticity computed from the velocity flow field in Stage 2, and $\vec{\omega}$ is the vorticity to be found.

Minimization of Equation 4 with respect to $\vec{\omega}_s$ means that both terms must remain as small as possible with respect to the L2 norm. The minimized sum (in brackets) represents the weak form of the VTE and enforces the physical flow structures in $\vec{\omega}_s$, while the term outside the brackets (i.e. $\|\vec{\omega} - \vec{\omega}_s\|_2^2$) links $\vec{\omega}$ and $\vec{\omega}_s$ such that the difference in the L2 norm between these two vector fields is minimal. The balance between the two components dictates the strength of the restoration and this, in turn, is controlled by a control parameter, α that has the dimensions of time (discussed in Section 4). The weak form of the VTE reverts to the normal form of the VTE, after the first variation in $\vec{\omega}$ is computed.

The first variation of this functional is:

$$\vec{\omega} - \alpha\nu\triangle\vec{\omega} = \vec{\omega}_s - \alpha\vec{e}(\vec{v}_s) \tag{5}$$

Note that if $\vec{\omega}_s$ satisfies the VTE, $\vec{\omega} = \vec{\omega}_s$.

In cases where the exact boundary conditions are known, solving Equation 5 is easily done analytically or numerically. In all other cases, it is assumed that volume V freely allows in-/out-flow (i.e. it is open), requiring that constant-flux boundary conditions must be used:

$$\left.\frac{\partial \vec{\omega}}{\partial n^-}\right|_{\partial V_l} = \left.\frac{\partial \vec{\omega}}{\partial n^+}\right|_{\partial V_l} \tag{6}$$

where, n^- is the inner normal to V and n^+ is the outer normal to V.

Such boundary conditions are sufficient in solving Equation 5 and do not rely on fixed vorticity or velocity fluxes. The filtered vorticity flow field $\vec{\omega}$ is then passed to Stage 4.

2.4 Stage 4: velocity restoration

The velocity restoration is done by minimising the functional:

$$\min_{\vec{u}} \left\{ \|\vec{u} - \vec{v}_s\|^2_\Omega + \|\nabla \times \vec{u} - \vec{\omega}\|^2_\Omega \right\}. \tag{7}$$

This is implemented similarly to Equation 4, and the output is an optimum velocity flow field, u, determined from Stage 2 and Stage 3. Here, term $\|\vec{u} - \vec{v}_s\|^2_\Omega$ links the output u and velocity field v_s from Stage 2 such that the L2 norm difference between them is minimal (and therefore also the experimental measurements), while the term $\|\nabla \times \vec{u} - \vec{\omega}\|_{\Omega}^{2}$ links the output pattern of the velocity flow field in u and the restored pattern of the vorticity flow field in $\vec{\omega}$ from Stage 3. Dimensional consistency is achieved using a constant that equals one, but has the dimensions of length squared. For the sake of simplicity, this constant is omitted in later derivations.

The first variation of this functional is:

$$\vec{u} - \triangle\vec{u} = \vec{v}_s - \nabla \times \vec{\omega} \tag{8}$$

The boundary conditions to Equation 8 are the same as in Stage 3, and solving results in the rectified velocity flow field, $\vec{u}$.

Note that Equation 2, Equation 5 and Equation 8 each represent a low-pass filter that causes a suppression of energy that must be recovered. Although this suppression is negligible for a single iteration, it becomes considerable if the algorithm executes more than 10 iterations. Here, it is assumed that the main fraction of the noise energy present in the data collected is concentrated in the middle and high frequency part of the spectrum (e.g. white noise). Therefore, low-pass filtering causes the large decay of that fraction after the first iteration, while the decay of the true signal is insignificant. The implication of this is that, after the first iteration, the energy of the remaining low frequency part is negligible compared to the true energy of the flow, such that the energy of the noisy flow approximately equals the true energy of the flow. The energy of this flow is recovered starting from the second iteration when the output $\vec{u}$ is multiplied by the ratio between the energy of the first iteration and that of the rectified data.

2.5 Algorithm termination

Algorithm termination occurs after a user-predefined maximum number of iterations or when the mean angle deviation between u and v_m is less than user specified tolerance. If this is not met, the velocity flow field, u, is defined as if it were v_m and the process repeats using the results generated in the last output.

3 Algorithm development

Vlasenko and Schnorr (2010) established that PCEVD offers good restoration capabilities for any type of flow, corrupted by any type of noise. It is also able to accommodate gappy data, however the quality of this output is detrimentally affected by the sparsity. The large gaps within the velocity flow field are not considered as noise, as they meet the divergence-free criteria (Stage 2) and the trivial solution of the VTE (Stage 3 and Stage 4). Therefore, PCEVD merges the large gaps with the PIV or PTV data, changing the complete pattern of the velocity flow field. It is this failing especially, rather than the hydrodynamical theory applied, that prompted the development of a new algorithm, PEFRA. This new algorithm is applicable to any type of (incompressible) flow, and offers similar restoration capabilities to its PCEVD predecessor, but with less sensitivity to the sparsity of the data.

PEFRA consists of three blocks: interpolation, linear approximation and restoration. Here, weighted-average interpolation methods are used to infill gappy data in the first block. This is then smoothed by linearization, using a modified PCEVD algorithm (with Stage 2 omitted and $\vec{e}(\vec{v})$ in Stage 3 set to zero), such that it fits the pattern of the laminar vorticity flow field. Finally, restoration is done using a differently modified PCVED algorithm (with Stage 2 omitted) and the output velocity flow field established iteratively, as in §2. The omission of Stage 2 from PEFRA may be justified by its small effect on the reconstruction of gappy elements within the velocity flow field. The reason for this is that both Block 2 and Block 3 decrease the vorticity (proof in Appendix) on each loop, such that the output vectors are almost divergence-free. The scheme and pseudo-code of PEFRA for its numerical implementation are given in Appendix B.

3.1 PEFRA volume and boundary conditions

In cases where the boundary conditions are not known, continuity flux boundary conditions are used in both PEFRA and PCEVD. In PCEVD, these are applied to the same volume as that where the data were collected but, in PEFRA, a larger volume is needed. This is apparent when Equation 5 is considered, with respect of the normal vorticity component, at the boundary of V. These continuity flux boundary conditions convert Equation 5 to:

$$\vec{\omega}^n = \vec{\omega}_s^n - \alpha \vec{e}^n(\vec{v}_s). \tag{9}$$

where, n is the normal component of the vector.

Therefore, the unknown vorticity component, $\vec{\omega}$, is unambiguously defined by the difference between $\vec{\omega}_s$ and $\alpha\vec{e}(\vec{v}_s)$, where the noisy $\vec{\omega}_s$ is corrected by $\alpha\vec{e}(\vec{v}_s)$. However, when experimental measurements are highly sparse, Equation 9 is not appropriate as the lack of velocity data at the boundary means the fluxes in Equation 9 are computed incorrectly. Note that after interpolation and linearization, $\vec{v}_s$ is a linear function, as is $\vec{\omega}$ and $\alpha\vec{e}(\vec{v}_s)$. Consequently, ω is also linear – irrespective of the dynamics within the sample volume – requiring enlargement of this volume in PEFRA.

To understand these, a volume, V, containing the fluid motion, surrounded by a larger volume V_l of the same shape, is considered. The walls of

V and V_l are invisible to fluid movement and freely allow in-/out-flow. Critically, the center of these volumes are co-positioned, meaning the distance, d, that offset the walls of V from the walls of V_l are the same to each face. Therefore, if V_l is sufficiently large, any turbulence present in V diminishes at the boundary of V_l due to viscosity effects. Here, flows near the boundary are linear, so constant-flux boundary conditions (Equation 6) are appropriate.

To explain the computation of d, the analogy of fractal turbulence may be considered. Here, it is suggested that a velocity flow field may be represented as an overlapping set of vortices with different characteristic length scales (Giacomazzi et al., 1999). Let L be the characteristic length of the largest vortices in the set. Following Kolmogorov theory (Landau and Lifshitz, 2000), an individual eddy is divided into several vortices twice as small as the original after a distance of twice its characteristic length. Therefore, the largest vortices in the set are divided into several smaller vortices with a characteristic length of $L/2$ after a distance of $2L$. These smaller vortices are then sub-divided after a distance of L and the process repeats until the minimum eddy length scales are met. In discrete cases, this is set by the number of grid-points that are needed for the resolution of the smallest vortices (i.e. three grid-points). The equation for the minimum length of d is, therefore:

$$d = \sum_{i=0}^{N} \frac{L}{2^{i-1}}, \qquad N = \log_2 \left(\frac{L}{3} \right) \tag{10}$$

The enlargement of V to V_l by d means that flow near the boundary are constant and linear, so constant-flux boundary conditions (Equation 6) are appropriate. To emphasize that constant flux boundary conditions are applied to a larger volume where the pattern of the vorticity flow field is linear, these are termed open boundary conditions. If L is unknown, and estimation of d using Equation 10 is impossible, then this is able to be obtained iteratively. The algorithm to do so is as follows: initially, all control parameters are set as default (§4.3.1) and $d = 1$. PEFRA runs with this set of control parameters until the termination criterion is satisfied, and the root-mean-difference between the input and output velocity flow field is saved for further reference. Then d is incremented by one and the procedure repeated, whereupon the root-mean-square differences between the experimental measurements and the restored data from the present and the preceding iterations are compared. If the relative difference between these two values is sufficiently small (e.g. smaller than 1%) the algorithm terminates and V_l is estimated. Otherwise, d is incremented by one and the sequence repeated again. Note that if this tolerance is set close to zero, the estimated d will be the same as in Equation 10.

3.2 Interpolation

After the enlargement of V to V_l, all empty grid-points in V are filled by interpolation of the experimental measurements, prior to the velocity flow field from V being extrapolated into V_l. Tests using different types of interpolation (i.e. nearest neighbour, splines and weighted-average) reveal that weighted-average schemes are most appropriate, since they achieve the best convergence rate of PEFRA. Consequently, these schemes are used in this algorithm. Here, it is assumed that all the available PIV or PTV data are presented on a regular grid (or projected from an irregular grid onto a regular grid), with a grid-step h. Each empty node is surrounded by a sphere of $2h$. If there are two or more measured velocity vectors in that sphere, a weighted average interpolation can be applied and the node is filled with the interpolated data. If not, the radius of the sphere is increased by h and the availability of measured velocity vectors is re-checked. If, again, there are less than two recorded velocity vectors the radius of the sphere increased until the amount of measured vectors within the sphere becomes greater than or equal to two. The weights for interpolation are set as the inverse distance from the node to the center of the sphere.

3.3 Linearization

In several cases, ramps are present at junctions between the infilled data and the recorded velocity flow field, however the smoothing of these ramps by Gaussian Filtering (Stage 1) may be insufficient at avoiding large non-linear $\vec{e}(\vec{v})$ terms at these junctions. Increasing the filter variance will strengthen the severity of the smoothing of these ramps but this, in turn, risks over-smoothing the pattern of the velocity flow field such that two adjacent vortices may be amalgamated into one and so must be avoided. This over- or under-smoothing is prevented by fitting the interpolated velocity flow field to the linear VTE, since the linear VTE does not have problematic non-linear terms and can filter-out the junctions as discussed below. Helpfully, this solution of the linear VTE is also the first-order (linear) approximation of the non-linear VTE. This solution is obtained by performing a single Gaussian filtering operation, prior to executing step 3 and step 4, sequentially, with the linear VTE, until the termination criterion is satisfied. Therefore, the algorithm establishes linear flow such that, among all the possible linear solutions, the difference in the L2 norm of the velocity and vorticity, with the corresponding $\vec{\omega}_s$ and $\vec{v}_s$, is minimal. The energy of the flow is subsequently recovered, as in PCEVD. After each iteration, the obtained linear velocity field fills the gaps in the measurements. The resultant field is used then as an input field for the next iteration.

Note that PEFRA is an iterative method, and therefore its computational speed performance may be significantly improved if the correct initial estimate (known also as initial guess) is found. Since the linear flow is traditionally used as the first approximation of any type of flow (Pedlosky, 1990), the construction of linear flow is the preparation of this estimate. It decreases the time needed for the restoration in the final block – irrespective of the dynamics within the sample volume.

3.4 Restoration

The final block, restoration, consists of two stages. Initially, it is the same as linearization but with the full form of $\vec{e}(\vec{v})$ used for the vorticity restoration. Here, on each iteration, the grid-points containing the restored data are substituted with the non-zero data from the sparse experimental measurements. After the algorithm termination criteria is met, this last stage is again repeated only without the input of the PIV or PTV data into the output velocity flow field such that noise injected with the experimental measurements is filtered out. The energy of the flow is subsequently recovered, as in PCEVD.

4 Algorithm sensitivity

The sensitivity of PEFRA to noise, sparsity and control parameters is discussed analytically here, with an experimental verification provided in §5.

For the purposes of analysis, the restoration is considered to be successful if the L2 difference between the true flow and the restored flow decreases on each iteration, ultimately becoming less than a user-defined criterion. Although the true flow in experimental measurements is unknown, it is possible to anticipate the cases where restoration will be successful from only the characteristics of the PIV or PTV data. This is examined using an extreme example. Here, a velocity flow field only consisting of two vectors is considered. If the two vectors are far apart, then they may be connected to one large vortex or two smaller separate vortices (or, indeed, any other type of flow) and any later restoration will be ambiguous. Consequently, a necessary criterion for the successful restoration specifies that a velocity flow field fitting the PIV or PTV data must be unique. If this correct restoration is not still possible when any part of the velocity flow field is omitted then this flow is labelled as critically sparse. Therefore, this necessary criterion for the successful restoration is met if the sparsity of these data are above critical.

The necessary sparsity criterion for the successful restoration may be checked using homogeneously sparse velocity measurements, presented on a regular grid. Here, S is the sparsity of the data, i.e. the number of grid-points containing data, divided by the total number of grid-points (expressed in percent), while L_s is the characteristic length scale (expressed in grid-points) of the **smallest resolved**[1] entities within the measured, discrete, velocity flow field. According to §3, an approximation of the velocity flow field within the sample volume is yielded by an initial interpolation and subsequently improved and specified iteratively. The interpolation of the smallest entities of this flow is possible where at least two vectors are present at a distance of L_s, i.e. if the sparsity of the data satisfies a *critical sparsity condition*:

$$S \geq \frac{8}{L_s^3} \times 100\% \qquad (11)$$

In cases of turbulence, the number of grid-points that are needed for the resolution of the smallest vortices is four grid-points, meaning that for the correct restoration $S \geq 12.5\%$. It is suggested that 12.5% is considered to be the default value for critical sparsity, since all types of flows with $S \geq 12.5\%$ may be successfully reconstructed, providing the noise level in the experimental measurements is below its critical value (discussed below).

[1]The flow feature is resolved on the grid if all its velocity maxima and minima can be projected on the corresponding grid nodes

4.1 Algorithm sensitivity to noise (critically-sparse velocity flow field)

The sensitivity of PEFRA to a critically sparse velocity flow field containing noise, $\vec{\delta^o}$, is considered in reference to Equation 4. If the restoration of the pattern of the vorticity flow field is unaffected by noise, the only solution to this expression is the true vorticity, $\vec{\omega^T}$. The substitution of $\vec{\omega^T}$ into Equation 4 reduces term 1 to $\|\vec{\delta^o}\|$ and term 2 disappears. If this is affected by noise, the restoration results in a new vorticity flow field, $\vec{\omega^T} + \vec{\theta}$, where $\vec{\theta}$ is the difference between $\vec{\omega^T}$ and the new output. Since the output satisfies the VTE, the substitution of $\vec{\omega^T} + \vec{\theta}$ into Equation 4 reduces term 1 to $\|\vec{\delta^o} - \vec{\theta}\|$ and term 2 disappears. If this is minimized by $\vec{\omega^T} + \vec{\theta}$ it must be true that:

$$\frac{J(\vec{\omega^T})}{J(\vec{\omega^T} + \vec{\theta})} = \frac{\|\vec{\delta^o}\|_\Omega^2}{\|\vec{\delta^o} - \vec{\theta}\|_\Omega^2} > 1 \tag{12}$$

The inequality on the RHS of Equation 12 is true if $|\vec{\theta}| < 2|\vec{\delta^o}|$, meaning that if the extremely sparse velocity measurements contain 5% noise, the difference between the true vorticity and the post-restoration vorticity is less than 10%. Therefore, the critically sparse velocity flow field will be successfully reconstructed, with data containing much less than 50 % of the noise, i.e.:

$$\frac{\|\vec{\delta^o}\|_\Omega^2}{\|\vec{\omega^T}\|_\Omega^2} \ll 0.5 \tag{13}$$

Note that Equation 13 considerably underestimates the upper limit of the noise level in the input data permissible for successful restoration to still be achieved. In reality, successful restoration is possible even when $\|\vec{\delta^o}\|_\Omega^2/\|\vec{\omega^T}\|_\Omega^2 \simeq 0.5$., however as Equation 13 unambiguously ensures successful restoration, it is this that is used for the noise level condition.

4.2 Algorithm sensitivity to noise (non critically-sparse velocity flow field)

The sensitivity of PEFRA to a non-critically sparse velocity flow field is identical to that completed for the PCEVD algorithm (cf. Vlasenko 2010, where a detailed study of the effect of noise in the data at each restoration stage of the algorithm is presented). Since PCEVD and PEFRA are from the same pedigree, these conclusions will remain the same for the present algorithm, so only a summary is provided here.

According to Vlasenko (2010), the noise in the experimental measurements contains a fraction that satisfies the VTE and, consequently, will be referred to here as the hydrodynamical component of the noise. Therefore, the velocity estimates generated from noisy PIV or PTV data, f, may be considered as consisting of the sum of three components: $f = \vec{v}^T + (\vec{h} + \vec{\vec{\delta}})$, where $\vec{v}^T$ is the true velocity, and the expression in brackets is noise consisting of a hydrodynamical component ($\vec{h}$) and a non-hydrodynamical component ($\vec{\delta}$), that does not satisfy VTE. The algorithm sensitivity to each of these is considered separately below.

4.2.1 The hydrodynamical component of the noise

The hydrodynamical component of the noise is a systematic error of both PCEVD and PEFRA that cannot be eliminated. The results will therefore be identical to that established for the earlier algorithm. Vlasenko (2010) applied PCEVD to two sets of data, each of 1000 vector fields, consisting of pure identically-distributed white noise with zero-mean and pure Gaussian-distributed white noise with zero-mean, respectively. These data suggest that if the noise contain such a component, it will pass the PCEVD filtering. Therefore, the application of PCEVD to these data revealed that each of the 1000 vector fields in the two sets contain a pattern suggestive of a turbulent motion, whose substitution into the discrete VTE results in equality. Figure 1 is an example of one of these vector fields, obtained from one of the 1000 samples of white noise. It was established that in the two sets, the fraction of the hydrodynamical component of the noise obeys the same bell-shaped distribution. Its mean, variance and maximum (normalized by the noise level) equals 0.115, 0.510 and 13, respectively. These experiments with both types of noise revealed that the hydrodynamical component of the noise always results in an arbitrary isotropic turbulent-like pattern (e.g. Figure 1) if the noise level in each component is identical. However, if the noise level in one component is significantly greater than for the others, it results in a flow field, satisfying the VTE, with anisotropy in that component. In cases of zero-mean distributed noise, the anisotropy causes a pattern similar to Kelvin-Helmholz instabilities. In cases of nonzero-mean distributed noise, the noise-pattern appears embedded within the constant background flow, whose components are proportional to the mean of the noise in the corresponding velocity components. Due to nonlinear terms, the VTE does not possess the property of linear additivity, meaning that if noise is present in measurements it will affect the form of the hydrodynamical component. These statistical experiments with artificial measurements revealed a weak anti-correlation, which is not smaller than -0.1. The subtraction of the corresponding artificial true velocity field from the restored output shows that, with the exception of differences in small details, the hydrodynamical component remains the same as the hydrodynamical component filtered from the pure noise. On the results of these experiments Vlasenko (2010) concluded that noise contains a hydrodynamical component that cannot be removed by PCEVD (nor by PEFRA) as it is merged with the output data. Defining n as the inverse of the signal-to-noise ratio (i.e. the ratio between the L2 norms of the noisy and true velocity flow field), the fraction of this component in the output is greater than $0.9n$ but less than $13n$ for zero mean noise. If the noise has nonzero mean, the hydrodynamical fraction is estimated as the sum of the mean noise level and $0.13n$.

4.2.2 The non-hydrodynamical component of the noise

If it is assumed that noise exists homogeneously within the sample volume and that this is able to be expanded spectrally, where a_i is the amplitude of these harmonics at a spatial frequency of $\phi = L/i$ $(i = 1, 2, ..., N)$ and U is defined as twice the characteristic velocity. According to Vlasenko (2010) an approximation of the non-hydrodynamical component of the noise is yielded by:

$$\epsilon_i \leq \underbrace{\exp^{-(\sigma i)^2/2}}_{1} \underbrace{\frac{a_i}{1+i^2}}_{2} \left(\sqrt{1 + \left(\underbrace{\frac{U}{(\phi^2 \alpha)^{-1} + \nu}}_{3} \right)} \right) \tag{14}$$

where, ϵ_i is the harmonics remaining after one iteration of the restoration in the final block. Term 1, term 2 and term 3 (in under-brackets) represent the eigen-reduction factors of the noise of the Gaussian filtering, vorticity and velocity restoration steps, as if these are applied independently. The upper bounds for the non-hydrodynamical component of the noise remaining in the data at each step (separately) are provided in Vlasenko (2010). Equation 14 is an approximation of the upper bound of the joint impact of these errors (from all stages) in the restoration block. This expression is, however, difficult to apply practically. A more convenient expression is achieved through correct selection of control parameters ν and α (§4.3). If this is done, the product of term 2 and the expression under the square-root in Equation 14 is less than or equal to one, and ϵ_i may be expressed as: $\epsilon_i \leq \exp^{-(\sigma)^2/2} a_i$. When the L2 norm is subtracted from the LHS and RHS and both, in turn, are divided by the L2 norm of the true velocity flow field, a new inequality (in terms of the signal-to-noise ratio) is yielded: $n_r \leq \exp^{-(\sigma)^2/2} n_n$, where n_n and n_r are the inverse of the signal-to-noise ratio of the non-hydrodynamical component of the noise before and after the restoration in turn. Since the non-hydrodynamical component of the noise is a fraction of the noise quantified by the inverse of the signal-to-noise ratio, n, i.e. $n_n \leq n$, then it must be true that: $n_r \leq \exp^{-(\sigma)^2/2} n$. Using this inequality and the estimates for the hydrodynamical component of the noise, the total error remaining after the restoration may be expressed as:

$$n_{total} \leq n(0.13 + \exp^{-(\sigma)^2/2}) \tag{15}$$

As an example, if $\sigma = 1.34$, then according to the inequality, $n_{total} \leq 1$, when $n = 2.2$. Similarly as in Equation 12, the inequality underestimates the upper limit of the noise level in the input data permissible for successful restoration to still be achieved.

4.3 Sensitivity to control parameters

The sensitivity of PEFRA to control parameters, σ, α and ν, is considered in reference to Equation 14. Term 1 is the error reduction from Gaussian filtering and is always less than one and, therefore, never causes an increase in the noise-level. In fact, the opposite is true as an increase (linearly) in parameter σ (§2) decreases the noise-level exponentially, as well as smoothing the pattern of the velocity flow field. However, to prevent over-smoothing, Vlasenko (2010) established that σ must be less than 1.34. Similarly, term 2 is the error reduction from velocity restoration and this is always less than one. This is affected by term 3, that characterizes the upper limit of the impact of the vorticity restoration on the velocity restoration. Since the term under the square root is always more than one, it is possible that $\epsilon_i > a_i$ and this, in turn, causes an increase in the noise-level. To ensure that this upper limit is not achieved $\epsilon_i/a_i < 1$ and the control parameters selected accordingly. When the left hand side and the right hand side of Equation 14 are divided by a_i, the right hand side is less than one. Simple mathematical operations show that this right hand side is always less than one if:

$$0 < \frac{U}{\alpha^{-1} - 3\nu} < 1 \tag{16}$$

Therefore, the permissible values of α and ν are unambiguously defined by Equation 16 (referred to as *nu-alpha condition*). Note that the spatial frequency in front of α^{-1} is set to one and omitted here. However, it is important to remember its dimensions ($\mathrm{m\,s^{-1}}$) remain and these balance the denominator.

4.3.1 Optimum selection of control parameters

If the nu-alpha condition is satisfied, the sparsity and quantity of noise in the data allow successful restoration, and the noise in the experimental measurements has a zero-mean, then the noisy velocity flow field and the reconstructed velocity fields may be expressed as: $\vec{v}_{noisy} = \vec{v}_{true} + \vec{N}$ and $\vec{v}_{PEFRA} = \vec{v}_{true} + \vec{A} + \vec{N}_h$. Here, $\vec{v}_{true}$ is the true velocity flow field, $\vec{N}$ is noise in the experimental measurements, $\vec{N}_h$ is the hydrodynamical component of $\vec{N}$ and $\vec{A}$ represents the artefacts caused by poor selection of control parameters. The residual between the noisy velocity vectors and the reconstructed velocity vectors at the grid node k is $\vec{v}^k_{noisy} - \vec{v}^k_{PEFRA} = \vec{N}^k - \vec{N}^k_h - \vec{A}^k$. According to §4.2.1, if $\vec{N}$ has a zero-mean, $\vec{N}_h$ has an arbitrary isotropic noise-pattern (and therefore the difference $\vec{N}' = \vec{N} - \vec{N}_h$ also has zero-mean), and

$\vec{v}^k_{noisy} - \vec{v}^k_{PEFRA} = \vec{N}'^k - \vec{A}^k$, the root-mean-square difference between the true velocity flow field and the reconstructed flow field may be estimated as:

$$\Delta = \sqrt{\frac{1}{K}\sum_k^K (\vec{v}^k_{noisy} - \vec{v}^k_{PEFRA})^2} = \sqrt{\overline{A^2} - 2\overline{A \cdot N'} + \overline{\vec{N}'^2}} \tag{17}$$

where the overline denotes averaging. Note that $\vec{N}'$ has no hydrodynamical component, which means that that $\vec{A}$ and $\vec{N}'$ are independent. Moreover, $\vec{N}'$ has zero mean, hence $\overline{\vec{A} \cdot N'} = \overline{\vec{A}} \cdot \overline{\vec{N}} = 0$. Equation 17 therefore may be simplified to:

$$\Delta = \sqrt{\frac{1}{K}\sum_k^K (\vec{v}^k_{noisy} - \vec{v}^k_{PEFRA})^2} = \sqrt{\overline{A^2} + (1-C)^2\overline{N^2}} \tag{18}$$

where $C \in [0.09, 0.13]$ is the fraction of hydrodynamical component in $\vec{N}$. If the noise in the experimental measurements has a nonzero mean, the reasoning and intermediate conclusions remain the same – only the data $\vec{A}$, $\vec{N}$ and $\vec{N}_h$, are expressed as the sum of the corresponding zero mean variables $\vec{A}_0$, $\vec{N}_0$, $\vec{N}_{0h}$ and their corresponding means. The root of the mean-square-difference may then be computed by repeating the reasoning above. Since the arithmetic for this is cumbersome, it is omitted here and the final expression is provided instead:

$$\Delta = \sqrt{\frac{1}{K}\sum_k^K (\vec{v}^k_{noisy} - \vec{v}^k_{PEFRA})^2} = \sqrt{\overline{A_0^2} + (1-C)^2\overline{N_0^2} + \mu^2} \tag{19}$$

where μ is the sum of means of $\vec{A}$ and $\vec{N}$. Note that Δ in Equation 18 and Equation 19 is minimal when $\overline{A^2}$ and $\overline{A_0^2}$ are minimal. The artefacts are, in turn, minimal only when the optimum set of parameters are selected. Therefore, the problem of finding of optimum set of parameters is equivalent to the problem of finding the set of parameters that minimize Δ.

The search of parameters that minimize Δ may be achieved, for example, using the gradient descent method (cf. Talagrand and Courtier 1987), with the following control parameters used by default for the computation of the first gradient step: $\sigma = 1.34$ (see Vlasenko and Schnorr (2010)), ν can be set to its physical value and $\alpha = (U^{-1} + 3\nu)^{-1}$, starting at the boundary of nu-alpha condition (Equation 16), where twice the maximum velocity of the noisy flow can be used as U. Note that if the noise in the experimental measurements is homogeneously distributed in both time and space, the control parameters may be considered the same for all frames. The simplest version of this algorithm is presented in the pseudo-code outline of PEFRA (Table 4 in Appendix B.

4.3.2 Estimation of maximum discrepancy between true and restored flows

An important corollary of §4.3.1 will occur under ideal conditions, where $\vec{v}^k_{PEFRA} = \vec{v}_{true}$, or where the experimental measurements are noise free, and

$\vec{v}^k_{noisy} = \vec{v}_{true}$. In these cases, Equation 19 is never equal to zero. Note that in noise free measurements $\Delta = \sqrt{\overline{\vec{A}^2_0} + \mu^2}$ measures only the fraction of artefacts in the restored data, while the occurrence of noise in data only causes an increase in Δ. Therefore, the root-mean-square difference between the **true** velocity flow field and **restored** velocity flow field never exceeds Δ. If the mean and the variance of $\vec{N}$ are known (e.g. from a reference experiment with constant flow), Equation 19 is an exact estimate of the root-mean-square difference between the true and restored velocity flow field.

4.4 Algorithm sensitivity to flow parameters: time, length, velocity.

Velocity Due to the assumption of incompressibility PEFFRA may only be applied to a flow where the Mach number is much smaller than one.

Length The quality of restoration for any individual flow entities depends on its grid-representative characteristic scale (expressed in grid-points) but not on its actual size. According to Vlasenko (2010), the energy spectrum of the rectified velocity flow field is proportional to $1/(1 + \nu\phi^2)$, where ϕ is a discrete frequency, inversely proportional to the characteristic length (expressed in grid-points). Following Kolmogorov theory, the high band part of the energy spectrum will obey the $-5/3$ law. Therefore, in cases of turbulent flow, the high-band part of the energy spectrum of the rectified velocity flow field is steeper than expected. As a consequence, the small-scaled (in terms of grid-scales) flow entities associated with high frequencies present in the rectified velocity flow field are always smoother than the same entities in the true velocity flow field. However, tests using the artificial data containing zero-sparsity, obtained from direct numerical simulations, revealed that this smoothing error – defined as mean-square-difference between the input and output velocity flow field – is of the order of 0.1%.

Time PEFRA uses the full VTE and therefore its accuracy in time depends only on how accurately the selected numerical scheme approximates the time derivative in the VTE. If τ is a time interval between two measurements, and O is big O notation, then for the first-order directed difference this error equals $O(\tau)$.

4.4.1 Summary of algorithm sensitivity to noise, sparsity and control parameters

In summary, successful restoration is possible for a critically sparse velocity flow field when Equation 13 is satisfied and for a non-critically sparse velocity flow field when Equation 15 is satisfied, and both the critical sparsity condition (Equation 11) and the nu-alpha condition (Equation 16) are met.

If the critical sparsity of the experimental measurements is not known, then 12.5% may be used by default. Equation 18 and Equation 19 estimate the maximum discrepancy between the true flow and the restored flow for the zero-mean and the non-zero mean noise respectively, while the minimization of Δ with respect to α, ν and σ yields the optimum set of parameters.

5 Algorithm performance

The performance of PEFRA is assessed using a series of twin-experiments, where the true velocity flow field is provided by Direct Numerical Simulation. From this artificial/numerical data, vectors are removed and noise added, such that a gappy and noisy sample is generated. After restoration, the results are compared to the true flow to establish if the two are similar (i.e. like"twins").

For these tests, direct numerical simulation data modelling turbulence in the wake of a cylinder (computed on a three-dimensional grid that consists of $128 \times 256 \times 128$ grid-points) and that of the development of a convection cell within a tank (that consists of $32 \times 32 \times 132$ grid-points) were used. The quality of the subsequent restoration is assessed normalized using the root-mean-square error, Δ_n, and the mean angle deviation, θ.

The Δ is defined as:

$$\Delta_n = \frac{\|\vec{v}_{true} - \vec{v}_{PEFRA}\|_2}{\|\vec{v}_{true}\|_2} \tag{20}$$

and measures the total difference between the true flow, $\vec{v}_{true}$, and the PEFRA output, $\vec{v}_{PEFRA}$. Note that Δ_n is the same as Δ discussed in §4.3.2, and $\vec{v}_{noisy} = \vec{v}_{true}$, but normalized using the root-mean-square of the true flow. For the twin experiments Δ_n is more convenient than Δ, since it measures the relative deviation of the restored flow from the true flow.

The θ is defined as:

$$\theta = \frac{\int_V |\arccos(\vec{v}_{true} - \vec{v}_{PEFRA})| d\mathbf{x}}{\int_V d\mathbf{x}} \tag{21}$$

and measures the mean angle difference between the true flow, $\vec{v}_{true}$, and the PEFRA output, $\vec{v}_{PEFRA}$. Therefore, if all the vectors in $\vec{v}_{PEFRA}$ have the same direction (i.e. the same pattern of the velocity flow field) as $\vec{v}_{true}$, then $\theta = 0$. Similar measures with $curl(\vec{v}_{true})$ and $curl(\vec{v}_{PEFRA})$ are used to qualify the vorticity reconstruction. They are denoted as Δ^{curl} and θ^{curl}

5.1 Sensitivity to sparsity, control parameters and type of flow

Experiment 1: Sensitivity to sparsity. The sensitivity of PEFRA to sparse, noise-free velocity measurements is assessed using artificial/numerical data modelling turbulence in the wake of a cylinder. Here, two conditions are considered, where the sparsity of the data, S (Equation 11), is 30% (i.e. $> 2.5\times$ critical sparsity) and 12.5% (i.e. = critical sparsity), respectively. A horizontal cross-section (HXS) of this flow is presented in Figure 2A, while the sparse (input) conditions are presented in Figure 2B and Figure 2C. The black dots represent empty grid-points. To facilitate a visual post-restoration assessment, the HXS of the true flow is repeated in Figure 3A, and the PEFRA output is presented in Figure 3B ($S = 30\%$) and Figure 3C ($S = 12.5\%$). Despite the sparsity of the PEFRA input, the restoration of the pattern of the velocity flow field is almost completely achieved in both cases, as confirmed by the quality statistics, where $\Delta_n = 0.1180$, and $\theta = 7.8860$, when $S = 30\%$ and $\Delta_n = 0.2260$, and $\theta = 11.2600$ when $S = 12.5\%$. A small difference between these two may be seen in fine details of the vorticity flow field, however the three-dimensional iso-surfaces of these both resemble the true flow. The iso-surfaces of vorticity absolute (further referred to as vorticity iso-surfaces) are used here for the visualisation of the reconstruction capabilites of PEFRA vorticism. The iso-surfaces in all experiments correspond to the mean of the true vorticity absolute. The vorticity iso-surface of the true flow is presented in Figure 4A, and the PEFRA output is presented in Figure 4B ($S = 30\%$) and Figure 4C ($S = 12.5\%$). The vorticity iso-surface of $S = 30\%$ is similar to the true flow, except in fine details such as the artificial tongue seen in the lower-left corner of Figure 4B. The artificial tongue also occurs in the vorticity iso-surface of $S = 12.5\%$, with it apparent the quality of the restoration decreases with the sparsity of the data (such that only large-scale components in Figure 4C resemble the true iso-surface in Figure 4A). The quality statistics show that when $S = 30\%$, $\Delta^{curl} = 0.2120$ and $\theta^{curl} = 12.43$ but when $S = 12.5\%$, $\Delta^{curl} = 0.4112$, and $\theta^{curl} = 20.680$.

Experiment 2: Sensitivity to sparsity and type of flow. To extend the analysis, the algorithm performance is assessed under different flow conditions (such as adjacent to a rigid boundary) using artificial/numerical data modelling the development of a convection cell in a tank. The sinking of the cold, dense fluid generates two vortices, each with a characteristic length equalling half the length of the tank (i.e. 16 grid-points). Therefore, the critical sparsity (Equation 11) of this flow is 98%. A vertical cross-section of this flow is presented in Figure 5A, while the sparse (input) conditions are presented in Figure 5B. The black dots again represent empty grid-points. To facilitate a visual post-restoration assessment, the vertical cross-section of the true flow is repeated in Figure 6A and the PEFRA output is presented in Figure 6B. Note that the tank has rigid walls, meaning that exact boundary conditions may be defined. However, these exact boundary conditions were not used in place of the constant flux conditions specified in §3, enabling their application to a velocity flow field bounded by rigid walls to be assessed. Again, the restoration of the velocity flow field is almost completely achieved, even at its edges, as confirmed by θ (11.9000°) being similar to that for the wake of the cylinder. Under these conditions, Δ_n (0.4200) for the convection cell is larger. Such a large difference in Δ_n and small difference in θ indicates that, in cases of critical sparsity, the restoration of the direction (pattern) of the vectors is independent of the type of flow, while their magnitude (length) is flow dependent. The reason for this dependency is that the mean lengths of these vectors are proportional to the square-root of the mean energy of the flow. Due to the filtering attributes of PEFRA (§2), the average energy of the PEFRA output decreases after every iteration. This is compensated by setting it to the average energy of the sparse velocity flow field as it is assumed these (sparse) non-zero vectors are a representative sample of the true flow, and therefore their average energy is also representative (§2). However, in cases of a small volume containing highly sparse velocity measurements, this sampling is not representative and PEFRA cannot correctly recover the energy. Increasing the sparsity of the data beyond the critical level causes the algorithm to fail completely. An example of this failure is seen in Figure 6C, where the sparsity is 99%. Therefore, Equation 11 permits a correct estimate of the sparsity bounds where successful restoration is possible.

Experiment 3: Sensitivity to control parameters. In Figure 2 and Figure 5, the optimum set of parameters were used to facilitate the restoration. For the example of the wake of the cylinder (Figure 2), $\nu = 0.0025$, $\sigma = 0.1000$ and $\alpha = 0.0025$. If σ and ν are too large, over-filtering results (§4.3). The effects of this over-filtering is presented in Figure 7, where the same flow as in Figure 2A ($S = 30\%$) is used where $\nu = 2$ (Figure 7A) and $\sigma = 2$ (Figure 7B). These parameters cause the small-scale velocity components to be amalgamated or over-smoothed. If, however, α is too large, the nu-alpha condition is violated and this, in turn, causes the redundant small-scale velocity components that are seen in Figure 7C (where $\alpha = 2$, i.e. $6.5\times$ higher than that permitted in Equation 16).

5.2 Sensitivity to sparsity and noise and comparison with other methods

Experiment 4: Sensitivity to noise (critically-sparse velocity flow field). The restoration capabilities of PEFRA under extreme conditions (i.e. both critical sparsity and high noise level) are assessed using numerical data of the wake of a cylinder, but from a different time-step to that considered earlier, where the sparsity of the data, S, is 12.5%. In addition, white Gaussian noise (signal-to-noise ratio = 2) is added such that the quality statistics for the resultant gappy and noisy velocity flow field are $\Delta_n = 1.0260$ and $\theta = 52.4800°$. The sparse conditions are illustrated by the vectors within a HXS (Figure 8A). The HXS of the true flow is presented in Figure 8B and its three-dimensional vorticity iso-surface presented in Figure 8C, such that they may be compared to the PEFRA outputs in Figure 9A and Figure 10A, respectively. Again, the difference in the quality statistics ($\Delta_n = 0.3230$ and $\theta = 20.9390°$, and $\Delta^{curl} = 0.5429$ and $\theta^{curl} = 26.9390°$) is seen in fine details, while the large-scale features still resemble the true flow. Note that from Equation 12, it is possible that $\Delta_n \sim 2$ however, after restoration, the remaining error in this flow is almost a factor of 2 less than in the gappy and noisy velocity flow field. This fact warrants a comment on Equation 12 that this noise reduction is possible even when the critically sparse velocity flow field is highly contaminated by noise. At the same time, θ decreases by almost a factor of 2.5. In the equivalent tests without noise ($S = 12.5\%$), Δ_n decreases by a factor of 2, while θ decreases by a factor of 1.5. Therefore, the error of the restoration of gappy and noisy data (with signal-to-noise ratio = 2) causes an increase in the error of the restoration by a factor of 2. Consequently, it is concluded this restoration is successful even if the velocity flow field is critically sparse and contaminated by noise.

Experiment 5: Comparison with other methods. To complement the assessment of the algorithm performance, PEFRA is compared to PCEVD and Weighed Average Interpolation (WAI). The connection to PCEVD is made to show the benefit of the new algorithm over its predecessor. The connection to WAI is made to facilitate benchmarking against other methods as using specialist restoration method (e.g. PCEVD) is only meaningful to those familiar with that method. WAI, however, is both commonly used and easy to implement, and therefore can be a reference restoration method with which PEFRA or any other restoration method are compared. Here, the same gappy and noisy velocity flow field presented in Figure 8A is processed using PCEVD (Figure 9B and Figure 10B) and WAI (Figure 9C and Figure 10C), respectively. It was established above that the same data was mostly recovered by PEFRA, as confirmed by the quality statistics, where $\Delta_n = 0.3230$ and $\theta = 20.9390°$. In contrast, the PCEVD output has little in common with the true flow and, consequently, $\Delta_n = 99.0000$ and $\theta = 87.0000°$, $\Delta^{curl} = 346.12$ and $\theta^{curl} = 102.03°$. The implication of this is that vectors are orientated randomly with respect to the true solution and the restoration failed completely. The WAI output is an improvement over PCEVD ($\Delta_n = 0.9130$ and $\theta = 43.969°$,$\Delta^{curl} = 1.132$ and $\theta = 56.7°$), however these input vectors are too gappy and too noisy for the pattern of the resultant velocity flow field to be easily identified.

Dependency of restoration performance on inhomogeneity The restoration performance is inversely proportional to the quantity of the hydrodynamical component of the noise and PEFRA artefacts remaining in the data. The difference between the true flow and restored flow yields a vector field which is a merger of the hydrodynamical error and PEFRA artefacts remaining in the restored data. Such a difference, presented as a vector field in Figure 11, is obtained for the flow represented in Figure 8A (experiment 4). The length of the vectors at each grid-point represents the magnitude of the error at that point, while its direction does not have any particular sense. Note that although the true flow and restored flow (see Figures 8B and 9A) exhibit an isotropic pattern in their center and an anisotropic pattern at their edges, the error still remains isotropic. The relative root-mean-square of this vector field equals $\Delta_n = 0.3230$. For the similar field, with $S = 12.5\%$ but in the absence of noise, Experiment 1 revealed that the quantity of PEFRA artefacts, A, in the restored velocity flow field equals 0.22. According to §4.2.1, the mean quantity of hydrodynamical components may be estimated as $0.11n = 0.22$, where $n = 2$ is the noise level in the experiment. If the PEFRA artefacts and the hydrodynamical component of the noise are independent, the root of the sum of the squares of these two will be approximately equal to Δ_n in this experiment, which is confirmed. Therefore, the affects of sparsity and noise on PEFRA restoration are independent.

6 Implementation with 3D-PTV

PEFRA was developed for the restoration of gappy and noisy velocity measurements where the arrangement of a standard laboratory PIV or PTV set-up is not possible. Here, the assessment of the algorithm performance is extended to submersible 3D-PTV measurements in ocean flows, i.e. using data collected in-situ under extreme conditions.

Presently, our employment of 3D-PTV is for the study of the three-dimensional turbulence characteristics of the bottom boundary layer of the coastal ocean (Nimmo-Smith, 2008). Unlike laboratory measurements, where small neutrally-buoyant particles are seeded within the flow, plankton and suspended sediments are used as tracers. The use of these arises from the impracticality of seeding the ocean with tracers, meaning that a reliance on naturally available seed material is essential (Bertuccioli et al., 1999). The uneven shape of these particles especially, scattered inhomogeneously within the sample volume, causes an increase in the noise level since it cannot always be assumed that they act as passive tracers of the velocity flow field. In these cases, using PEFRA is highly beneficial, and this application is discussed below.

As in §5, the quality of the subsequent restoration is assessed using the normalized root-mean square error, Δ_n, and the mean angle deviation, θ.

The only difference is in normalization – selected to be the root-mean-square of the noisy velocity flow field. Since the in-situ velocity flow field has an arbitrary turbulent pattern and the PIV or PTV instrumentation is directionally independent, it is assumed that the noise has zero-mean and its level in these experimental measurements is at least twice as small as the level of the signal. In these cases, the variation between the root-mean-square difference of the noisy and the true flow is not greater than 12% and may be considered as approximately equal. Therefore, as before, Δ_n estimates the approximate relative maximum deviation from the true flow, permitting estimation of the optimum set of parameters, as discussed in §4.3.1 and §4.3.2.

If it is assumed that the plankton and sediments used as tracers are equally distributed within the small, arbitrarily turbulent sample volume, the experimental measurements have approximately constant level of noise and sparsity throughout the time series with small biases around this constant. Similarly, as sampling was conducted over periods of less than half an hour, and the site itself was sheltered from surface effects, the background flow conditions were also approximately constant throughout data collection. This means that restored velocity flow fields will have the same quality with the same level of artefacts. According to §4.3.1 and §4.3.2 Δ_n equals the sum of the root-mean-square of the noise in the data and artefacts produced by PEFRA during restoration. Any bias in noise or artefacts causes the corresponding bias in Δ_n, that over a sufficiently long time series will exhibit a random bell shaped distribution with a narrow variance. Following the random value distribution theory, it is expected that most of Δ_n biases will not exceed the variance, while the probability that Δ_n biases considerably exceed this value is close to zero. Therefore, an anomalous increase of Δ_n may be interpreted as an inconsistency in PEFRA or an incorrect assumption of homogeneous noise distribution for the instantaneous flow field. To arbitrate in such cases, the additional data available from 3D-PTV becomes important, as these contain an image of each of the particles and may be checked when unexpected results are encountered (Nimmo-Smith, 2008). Following Adrian and Westerweel (2010), it is expected that a small, regular particle will behave more like an ideal tracer – and, therefore, contaminate the velocity flow field less – than a large, more irregular particle. In addition, in the ocean, a minority of these large tracers may also be mobile plankton capable of independent movement. Consequently, the vectors established from tracking a small particle will need less adjustment by PEFRA, while the vectors established from tracking a large particle will need more adjustment by PEFRA. Therefore, if an instantaneous flow field is associated with an anomalous velocity arising from the presence of extremely large particles (or a high total number of large particles), it will be concluded that it is as a result of these tracers that the velocity flow field will contain more noise that results in an increase in Δ_n and θ. Moreover, it will be concluded that this is the only reason for the increase, and there is no inconsistency in PEFRA if the corrections of velocity vectors corresponding to small particles are much smaller than the corrections of velocity vectors corresponding to large particles.

6.1 Instrumentation

The submersible 3D-PTV system is detailed fully by Nimmo-Smith (2008). It consists of four 1002×1004 pixel 8-bit digital cameras that view a $20 \times 20 \times 20\,\mathrm{cm}^3$ sample volume illuminated by four 500 W underwater lights. Electrical power is supplied from a surface support vessel using an umbilical cable. The cable also enables communication with the 3D-PTV master computer, that synchronises the triggering of the cameras at the rate of 25 Hz. Data from each of these cameras is recorded by its own computer, each with 2×400 GB of hard disk storage (3.2 TB total). All underwater components are mounted on a rigid frame. A vane attached to the frame aligns it at an angle to the mean flow to prevent the contamination of the sample volume by the wake of the system. This alignment is monitored by an Acoustic Döppler Velocimeter (ADV) that also offers auxiliary turbulence statistics at the same height as the sample volume.

6.2 Data processing and use of PEFRA

After the calibration of the system (Svoboda et al., 2005), data processing is completed in three stages using the specialist 'Particle Tracking Velocimetry' software developed by Maas et al. (1993) and Willneff (2003). Here, particles are identified within the exposures from the four cameras by high-pass filtering, segmentation and weighted-centroid methods. In addition, maximum and minimum size criteria are used to limit contamination by noise or large objects. The calibration parameters are then used to relate the exposures from the four independent cameras, such that the three-dimensional position of the particles is yielded. Finally, tracking is done in image- and object-space, running the sequence in both directions so that linkages between adjacent frames are maximised, and the velocity of each of the particles at each time-step established by low-pass filtering their trajectories using a moving cubic spline (Luthi et al., 2005).

The experimental measurements are projected from an irregular grid onto a regular grid, where only the nearest neighbour of each of the detected particles are filled by interpolation (and all others set to zero) to minimise noise that arises from gridding. Similarly, if the distance, D, between each of the particles and the nearest grid node exceeds $0.5\sqrt{h_x^2 + h_y^2 + h_z^2}$ (where, h_x, h_y and h_z are the spatial discretization in X, Y and Z, respectively), these grid-points are set to zero also. Note that this algorithm is therefore adaptable to processor speed and memory such that, in theory, at an infinite resolution, all the particles will fall on the grid exactly.

6.3 *In situ* 3D-PTV experiments

The submersible 3D-PTV system was deployed on the east side of Plymouth Sound, Plymouth, UK, on 9 June 2005 in 12 m deep water on an ebb tide over a period of about 4 hours. The centre of the sample volume was set at the height of 0.64 m above the seabed. Data was recorded in 20 minute runs directly to hard disk storage.

For the following discussion, a right-handed Cartesian co-ordinate system is used, where X is aligned with the along-stream velocity component (U), Y is aligned with the cross-stream velocity component (V), and Z is aligned (upwards) with the wall-normal velocity component (W). Within this frame of reference, the zero-mean velocity is established using Reynold's Decomposition, i.e.:

$$u \equiv U - \langle U \rangle, \qquad v \equiv V - \langle V \rangle, \qquad \text{and} \qquad w \equiv W - \langle W \rangle, \tag{22}$$

where, $\langle \rangle$ is the mean of that velocity component.

Consistent with past in situ 2D-PIV measurements (Nimmo-Smith et al., 2002, 2005), a variety of different conditions were recorded, as characterised by different turbulence strengths ($I = \sqrt{u^2 + v^2 + w^2}$). Here, the restoration of two different conditions – corresponding to the 5th ($I = 0.6065$) and the 85th ($I = 1.0929$) percentile of the turbulence strengths during an example 10 minute time-series – are discussed. The sparsity of these flows are 2.14 % and 1.95 % while their characteristic lengths are 9 and 8 grid-points, in turn. Therefore, following Equation 11, the critical sparsity equals 1.09 % where $I = 0.6065$ and 1.56 % where $I = 1.0929$. Since the sparsity of these data exceeds the critical sparsity condition, it is expected that a successful restoration is possible.

Three orthogonal cross-sections of these flows are presented in Figure 12A to Figure 12C and Figure 12D to Figure 12F. The vectors corresponding to the PEFRA input (red) and the PEFRA output (black) are overlapped to illustrate the adjustment made. The projection of the convex hull of the tracked particles, representing the area where data were recorded, is shaded white. The subsequent restoration of these data culminates in the vorticity iso-surfaces presented in Figure 13A and Figure 13B. Qualitatively, Figure 13A exhibits small velocity gradients typical of a low turbulence level and Figure 13B is consistent with that expected of a higher turbulence level. While these cannot themselves confirm a correct restoration, the excellent agreement between the PEFRA input and the PEFRA output for the two different conditions, as well as that of the coherent structures and the turbulence level (Adrian, 2007), implies the physics of these flows have been successfully restored. Specific details of the restoration of Figure 13A and Figure 13B are quantified below.

Figure 14 presents an instantaneous velocity flow field where $I = 0.6065$. Here, 79 particles output by the tracking software survived filtering by moving cubic spline (Figure 14A). For the grid used ($h_x = h_y = h_z = 1\,\mathrm{cm}$), $D > 0.87\,\mathrm{cm}$ at one of these grid-points (red '+' markers). The interpolation of the velocity components onto the remaining grid-points results in a usable number of seed-points for the new algorithm of 78 (green '+' markers). After the application of PEFRA Δ_n and θ are quantified on a particle-by-particle basis (Figure 14B). The corresponding velocity flow field that has been modified by PEFRA is presented in Figure 14C, where the instantaneous sample volume mean velocity components have been subtracted from each of the vectors to reveal the three-dimensional turbulence structures. This is similar to the pattern of the velocity flow field presented in Figure 14D, where PEFRA was not applied. The cause of this similarity is that the sparsity of the data exceeds the critical sparsity condition by a factor of two and therefore will not affect the quality of the restoration. This, in turn, is aided by the small velocity gradients within the sample volume meaning that both large particles and small particles will follow the streamlines alike. Consequently, neither particles increase the noise level substantially.

Figure 15 presents an instantaneous velocity flow field where $I = 1.0929$. The format of these panels are the same as for the last figure, with 75 unique seed points used (Figure 15A). An increase in Δ_n and θ on a particle-by-particle basis (Figure 15B) is visible and more adjustment seen in the velocity flow field that was modified by PEFRA (Figure 15C) over that where PEFRA was not applied (Figure 15D). The cause of this adjustment is that the sparsity of the data is nearer the critical sparsity condition and therefore a very small part of this modification is likely to be an error (that increases as the sparsity of the data approaches the critical sparsity). This, in turn, is compounded by the large velocity gradients within the sample volume, as large particles cannot react to these as quickly as small particles and are affected by differential shear along their length.

As a verification of the adjustment made by PEFRA, the image containing a record of each of the particles must be examined to establish whether individual tracer characteristics (e.g. bubbles, large or heavy particles) are responsible for these differences. Figure 16 presents three sections of the image, viewed from each of the four different camera angles. The particles corresponding to the frame minimum Δ_n (0.6798) and frame minimum θ (0.0461) are highlighted in Figure 16A and Figure 16B. Although exhibiting the differences in shape expected of natural particles, these appear to be small in size and therefore the lack of adjustment is in agreement with the reasoning that they will not affect the noise level as much as a larger, more irregular particle. Accordingly, the particle corresponding to the frame maximum Δ_n (29.2589) and θ (15.9934) is revealed in Figure 16C to be a larger, irregular aggregate typical of a sediment floc. Such particles increase the noise level, and therefore need adjustment by PEFRA. Note that this connection to individual tracer characteristics is appropriate as there are a sufficient number of particles within the sample volume for the algorithm not to fail, while the small distance that separates these from their nearest grid-points (i.e. $D < 0.87\,\mathrm{cm}$) ensures that errors linked with interpolation will also be small.

This approach also provides a secondary method of validation. In 3D-PTV, individual particles are tracked as they are advected through the three-dimensional sample volume. If a time-series of the instantaneous velocity flow field is examined (Figure 17A, Figure 17B and Figure 17C), it may be seen from the stream ribbons that depict the gridded PEFRA output that the same coherent vortical structure is spatially and temporally coherent, and from the cones that depict the gridded particle positions that these progress through the sample volume. If the PEFRA output were incorrect, then there would be no coherence in the structure over the sequence of snapshots. Additionally, for any single particle moving through the sample volume, a similar correction (related to the individual tracer characteristics, as discussed with Figure 16) may be expected. Figure 17D and Figure 17E present a time-series the correction of a total of 12 different particles associated with the maximum and minimum adjustments that were made in Figure 17B to the total difference and angle deviation, respectively, over a sequence of 7 frames. These are seen to be both spatially and temporally invariant, giving confidence that it is the physical characteristics of the particles that causes the errors that are successfully corrected by PEFRA.

To complement the assessment of the instantaneous velocity flow fields presented above, Figure 18 shows a time-series of the particle and turbulence strength and total particle count (Figure 18A and Figure 18B), as well as the corresponding Δ_n and θ quantities (Figure 18C and Figure 18D). An increase in the sample volume mean turbulence intensities are generally connected to the passage of large coherent motions. This, in turn, is associated with the corresponding increase in Δ_n and θ that arises from tracking difficulties when the flow structures are more complex. In extreme instances of swimming particles not advected through the flow field, however, a single tracer can bias both restoration and turbulence statistics. An example of this is presented in Figure 19, where one particle is seen to move very differently to that of the pattern of the velocity flow field and necessitates a large adjustment by PEFRA (Figure 19A). The examination of the original image (Figure 19B) reveals that this 'particle' has a distinct body and tail, is 4.0 mm in length, and swims at a speed of 5.68 cm s^{-1}, or 14.2 body lengths per second. These quantities are consistent with laboratory measurements of the swimming speed of fish larvae (Bellwood and Fisher, 2001). This contamination is easily eliminated by removing single outliers using local Δ_n and θ anomalies and reprocessing the affected frame, but the example also confirms that PEFRA correctly identifies erroneous biological particles in situ.

7 Conclusions

A new Physics-Enabled Flow Restoration Algorithm (PEFRA) has been developed for the restoration of gappy and noisy velocity measurements where a standard PTV or PIV laboratory set-up (e.g. concentration/size of the particles tracked) is not possible, and the boundary and initial conditions are not known *a priori*. Implemented as a black box approach, where no user-background in fluid dynamics is necessary, this is able to restore the physical structure of the flow from gappy and noisy data, in accordance with its hydrodynamical basis. In addition to the restoration of the velocity flow field, PEFRA also estimates the maximum possible deviation of the output from the true flow. A theoretical and numerical assessment of the algorithm sensitivity demonstrates its successful employment under different flow conditions. When applied to submersible 3D-PTV measurements from the bottom boundary layer of the coastal ocean, it is apparent that using PEFRA is beneficial in processing data collected under difficult conditions, such as where the number (and reliability) of tracer-particles is very sparse.

Acknowledgements E. C. C. Steele was funded by the School of Marine Science and Engineering, Plymouth University, Plymouth, UK. The method for the denoising of fluid flows was adapted under a visiting fellowship award from the Marine Institute (Plymouth University), Plymouth, UK. Development of the 3D-PTV system was funded by the Royal Society, the Nuffield Foundation and the UK Natural Environment Research Council. We are grateful to C. Bunney, E. Davies and D. Uren for assistance with the deployment of the 3D-PTV system.

Appendix A

Let $\mathbf{p}$ be a divergence-free vector function. Following Vlasenko (2010),

$$\mathbf{q} - \mathbf{a\Delta q} = \mathbf{p} \tag{23}$$

(with constant flux boundary conditions applied) will only have a divergence-free solution. Therefore, the vorticity restoration in PCEVD and PEFRA will only have a divergence-free output. The equation for the velocity restoration is similar, however, in PEFRA, $\mathbf{p}$ is divergent, since this is not eliminated in $\vec{v}_s$ by solenoidal projection. To estimate the divergence remaining in the reconstructed velocity flow field after one iteration, the *div* operator is applied to both the LHS and the RHS of Equation 8. In doing so, the divergence-free term $\nabla \times \vec{\omega}$ on the RHS of Equation 8 disappears and the equation transforms to:

$$u - \triangle u = f \tag{24}$$

where, $u = div(\vec{u})$ and $f = div(\vec{v}_s)$.

Expanding u and f in a trigonometrical Fourier series, and substituting them into Equation 24, achieves:

$$u_n + 4(\pi n/L)^2 u_n = f_n, \qquad n = 1, 2, ..., N \tag{25}$$

where, u_n and f_n is the amplitude of harmonic n and L is the horizontal scale of the sample volume, V, where the data were recorded. Simple arithmetical manipulation achieves:

$$u_n = \frac{f_n}{1 + 4(\pi n/L)^2} \tag{26}$$

After each iteration, the divergence in $\vec{u}$ reduces by at least a factor of $1/(1 + 4(\pi n/L)^2)$, such that, after iteration i, this is by a factor of $1/(1 + 4(\pi n/L)^2)^i$. Therefore, with an increase in i, the divergence in $\vec{u}$ decreases, becoming negligible after several iterations.

Appendix B

The three tables comprising Appendix 7 are a pseudo-code representation of PEFRA, that follows the form of the MATLAB code written by the authors. Table 1 is a wrapper to PEFRA, and referred to as the PEFRA software. It sets the boundary conditions, finds the optimum set of parameters and launches the PEFRA function. The only user input needed in this software is to set the desirable tolerance and the viscosity of the fluid. The software then loads the time series of N velocity measurements (line 4), calibrates the size of V_l (lines 5-12) and determines the optimum set of control parameters (line 14), initialising the restoration of the measurements in the time series (lines 15-17). Table 2 outlines the PEFRA function, responsible for the interpolation of the data to the empty grid-points in V and extrapolation of the data into V_l (line 5), obtaining the linear flow field (lines 6-13) and performing the final restoration (lines 14-21). Table 3 outlines the PCEVD function, used by the software as external function. The stages of this algorithm are the same as discussed in §2 with the only difference being that Step 2 (Solenoidal projection) is not applied. The 'cgs' function and 'speye' operator used are the Conjugate Gradients Squared Method and Sparse identity matrix operator, respectively, as included with a core MATLAB distribution. The algorithm for obtaining the optimum set of control parameters is presented in Table 4.

% **- - - !!!! PROGRAM PEFRA !!!! - - -**

% *values* ν*, tol(desirable tolerance) and* τ *must be specified by user*

$[\vec{U}^{t=1:N}]$ = get_time_series % *read velocity measurements*

$(\vec{U}) = (\vec{U}^{t=1,2})$ % *first pair of vector fields*

$[\nu, \alpha, \sigma, d]$ = Set_default_values($\vec{U}$)

% *Initialization with* $\sigma = 1.34$, $d = 1$, $\alpha = (U^{-1} + 3\nu)^{-1}$

do

$[\vec{V}_1]$ = function_PEFRA($\vec{U}, \nu, \alpha, \sigma, \tau, d$)

d = d+1

$[\vec{V}_2]$ = function_PEFRA($\vec{U}, \nu, \alpha, \sigma, \tau, d$)

$[term]$ = termination_criterion($\vec{V}_1, \vec{V}_2$) % *term = true, when* $\|\vec{V}_1 - \vec{V}_2\|_2 < tol$

While (term_criterion = false)

% *search of optimal* (ν, α, σ)

$[\nu, \alpha, \sigma]$ = gradient_descent($\nu, \alpha, \sigma, \vec{U}, d$)

for t = 1: N % go through the whole time series

$[\vec{V}]$ = function_PEFRA($\vec{U}^t, \nu, \alpha, \sigma, \tau, d$)

end - - - !!!! END OF PROGRAM PEFRA !!!! - - -

Table (1). A wrapper to PEFRA, which computes boundary conditions, optimal set of parameters and starts PEFRA for the given time series.

function $[\vec{V}]$ = function_PEFRA$(\vec{U}, \nu, \alpha, \sigma, \tau, d)$

V_l = Set_Vl(d,size($\vec{U}$)) % *Enlarge* $\vec{U}$ *by given d, Set volume* V_l

Interpolate values into empty nodes

$[\vec{V_l}]$ = Interpolation_and_Extrapolation$(\vec{V_l})$

do % *Get linear flow*

$[\vec{V_l^k}]$ = function_Linear_PCEVD$(\vec{V_l}, \nu, \alpha, \sigma, \tau)$

% *In function_Linear_PCEVD, function Vector_E is substituted with* $\partial\vec{\omega}_s/\partial t$*,*

$[term]$ =termination_criterion$(\vec{V_l^k}, \vec{V_l^{k-1}})$ % *term = true, when* $\|\vec{V_l^k} - \vec{V_l^{k-1}}\|_2 < tol$

$k = k + 1$

$\vec{V_l} = \vec{V_l^k}$

$[\vec{V_l}]$ = inserter$(\vec{V_l}, \vec{U})$ % *Inserts nonempty values* $\vec{U}$ *into* $\vec{V_l}$

While (term_criterion = false)

do

$[\vec{V^k}]$ = function_PCEVD$(\vec{V_l}, \nu, \alpha, \sigma, \tau)$

$[term]$ =termination_criterion$(\vec{V_l^k}, \vec{V_l^{k-1}})$

$k = k + 1$

$\vec{V_l} = \vec{V_l^k}$

$[\vec{V_l}]$ = inserter$(\vec{V_l}, \vec{U})$ % *Inserts nonempty values* $\vec{U}$ *into* $\vec{V_l}$

While (term_criterion = false)

$[\vec{V_l}]$ = function_PCEVD$(\vec{V_l}, \nu, \alpha, \sigma, \tau)$ % *Final filtering*

Table (2). Function PEFRA.

function $[\vec{V}]$ = function_PCEVD$(\vec{U}, \nu, \alpha, \sigma, \tau)$ % *Without* **Step 2**

$\vec{U_s}$ = Gaussian_filter$(\vec{U}, \sigma)$ % - - - - - - **Step 1**
$\vec{\omega}_s$ = curl$(\vec{U_s})$
$\vec{e}$ = Vector_E$(\vec{U_s}, \vec{\omega}_s, \tau)$ % *vector_E computes LHS of VTE*

$\vec{F} = \vec{\omega}_s - \alpha\vec{e}$
A = speye(V_{lg}, V_{lg})-$\alpha * \nu$*Lap
% *Lap = Laplace operator in matrix form,* V_{lg} *= number of grid nodes in* V_l
$\vec{\omega}$ = cgs(A,$\vec{F}$) % - - - - - - **Step 3**
% it cgs = Conjugate Gradients Squared Method
B = speye(V_{lg}, V_{lg})-Lap
$\vec{F_2}$ = curl$(\vec{\omega})$+$\vec{U_s}$
$\vec{V}$= cgs(B,$\vec{F_2}$) % - - - - - - **Step 4**
$\vec{V}$= Energy$(\vec{U}, \vec{V})$% *Energy recovery*

Table (3). Function PCEVD.

1	function $[\vec{V}]$ = gradient_decent$(\vec{U}, \vec{V}, \nu, \alpha, \sigma, \tau, d)$
2	step = 0.05*σ; k = 1; $\Delta^1 = \infty$
3	**do**
4	$\Delta^{old} = \Delta^k$
5	$[\vec{V}]$ = function_PEFRA$(\vec{U}, \nu, \alpha, \sigma, \tau, d)$
6	Δ^k = delta_est$(\vec{U}, \vec{V})$ *compute* Δ *using Equation* (19)
7	k = k+1
9	**while(**$\Delta^{old} > \Delta^k + tol^{gr}$ **or** $k \leq 5$ **)** % *by default* $tol^{gr} = 0.001\Delta^{old}$
10	*repeat lines 2-9 for* ν *and* α
11	*if* $(, \nu, \alpha, \sigma, \tau)$ *is optimal, do all again until* $\Delta^{old} - \Delta^k < tol$

Table (4). The search of optimal set of parameters for PEFRA based on gradient descent method.

References

R. J. Adrian. Hairpin vortex organisation in wall turbulence. *Physics of Fluids*, 19:041301, 2007.

R. J. Adrian and J. Westerweel. *Particle image velocimetry*. Cambridge University Press, 2010.

D. R. Bellwood and R. Fisher. Relative swimming speed in reef fish larvae. *Marine Ecology Progress Series*, 2001.

L. Bertuccioli, G. I. Roth, J. Katz, and T. R. Osborn. A submersible particle image velocimetry system for turbulence measurements in the bottom boundary layer. *Journal of Atmospheric and Oceanic Technology*, 16:1635–1646, 1999.

A. Cenedese and G. Querzoli. Lagrangian statistics and transilient matrix measurements by PTV in a convective boundary layer. *Meas. Sci. Technol.*, 8:057002, 1997.

A. Cenedese and G. Querzoli. Particle tracking velocimetry with new algorithms. *Meas. Sci. Technol.*, 11:603–613, 2000.

J. Duncan, D. Dabiri, J. Hove, and M. Gharib. Universal outlier detection for particle image velocimetry (piv) & particle tracking velocimetry (ptv) data. *Measurement Science and Technology*, 21:1553–1561, 2010.

E. Giacomazzi, C. Bruno, and B. Favini. Fractal modelling of turbulent mixing. *Combustion Theory Modelling*, 3:637–655, 1999.

H. Gunes and U. Rist. On the use of kriging for enhanced data reconstruction in a separated transitional flat-plate boundary layer. *Physics of Fluids*, 20: 104–109, 2008.

D. P. Hart. PIV error correction. *Experiments in Fluids*, 29:13–22, 2000.

L. D. Landau and E. M. Lifshitz. *Fluid mechanics*. Butterworth-Heinemann, 2000.

B. Luthi, A. Tsinober, and W. Kinzelbach. Lagrangian measurement of vorticity dynamics in turbulent flow. *Journal of Fluid Mechanics*, 528: 87–118, 2005.

H. G. Maas, A. Grun, and D. Papantoniou. Particle tracking in three dimensional turbulent flows. part 1: Photogrammetric determination of particle coordinates. *Experiments in Fluids*, 15:133–146, 1993.

W. A. M. Nimmo-Smith. A submersible three-dimensional particle tracking velocimetry system for flow visualization in the coastal ocean. *Limnology and Oceanography: Methods*, 6:96–104, 2008.

W. A. M. Nimmo-Smith, P. Atsavapranee, J. Katz, and T. R. Osborn. PIV measurements in the bottom boundary layer of the coastal ocean. *Experiments in Fluids*, 33:962–971, 2002.

W. A. M. Nimmo-Smith, J. Katz, and T. R. Osborn. On the structure of turbulence in the bottom boundary layer of the coastal ocean. *Journal of Physical Oceanography*, 35:72–93, 2005.

T. Okuno, Y. Sugii, and S. Nishio. Image measurement of flow flow field using physics-based dynamic model. *Measurement Science and Technology*, 11: 667–676, 2000.

J. Pedlosky. *Geophysical fluid dynamics*. Springer-Verlag, 1990.

M. Raffel, C. E. Willert, S. T. Wereley, and J. Kompenhans. *Particle image velocimetry: a practical guide*. Springer, 2007.

A. Sciacchitano, R. P. Dwight, and F. Scarano. Navier–stokes simulations in gappy piv data. *Experiments in Fluids*, 29:1421–1435, 2012.

M. Stanislas, J. Westerweel, and J. Kompenhans, editors. *Particle image velocimetry: recent improvements*. Springer, 2004.

H. Stuer and S. Blaser. Interpolation of scattered 3d ptv data to a regular grid. *Flow, Turbulence and Combustion*, 64:215–232, 2000.

T. Suzuki. Reduced-order kalman-filtered hybrid simulation combining particle tracking velocimetry and direct numerical simulation. *Journal of Fluid Mechanics*, 709:249–288, 2012.

T. Suzuki, J. Hui, and F. Yamamoto. Unsteady ptv velocity field past an airfoil solved with dns: Part 2. validation and application at reynolds numbers up to re 104. *Experiments in Fluids*, 47:977–994, 2009a.

T. Suzuki, H. Ji, and F. Yamamoto. Unsteady ptv velocity field past an airfoil solved with dns: Part 1. algorithm of hybrid simulation and hybrid velocity field at re 103. *Experiments in Fluids*, 47:957–976, 2009b.

T. Svoboda, D. Martinec, and T. Pajdla. A convenient multi-camera self-calibration for virtual environments. *PRESENCE: teleoperators and virtual environments*, 14:407–422, 2005.

O. Talagrand and P. Courtier. Variational assimilation of meteorological observations with adjoint vorticity equation. i: Theory. *Quarterly Journal of the Royal Meteorological Society*, 113:1311–1328, 1987.

D. Venturi and G. Karniadakis. Gappy data and reconstruction procedures for flow past a cylinder. *Journal of Fluid Mechanics*, 509:315–336, 2004.

A. Vlasenko. *Physics-based fluid flow restoration method.* PhD thesis, University of Heidelberg, 2010.

A. Vlasenko and C. Schnorr. Physically consistent and efficient variational denoising of image fluid flow estimates. *IEEE Transactions on Image Processing*, 19:586–595, 2010.

J. Westerweel. Efficient detection of spurious vectors in particle image velocimetry data. *Experiments in Fluids*, 12:236–247, 1994.

J. Westerweel and F. Scarano. Universal outlier detection for PIV data. *Experiments in Fluids*, 39:1096–1100, 2005.

J. Willneff. *A spatio-temporal matching algorithm for 3D particle tracking velocimetry*. PhD thesis, ETH Zurich, 2003.

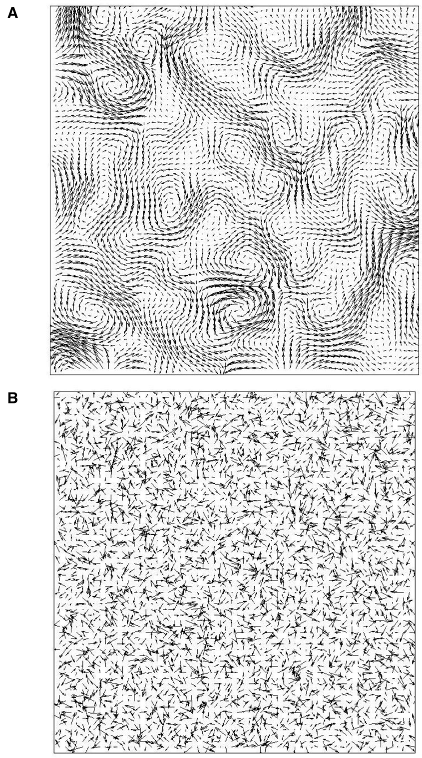

Figure (1). (A) The hydrodynamical component of noise, extracted from (B) the distribution of white Gaussian noise.

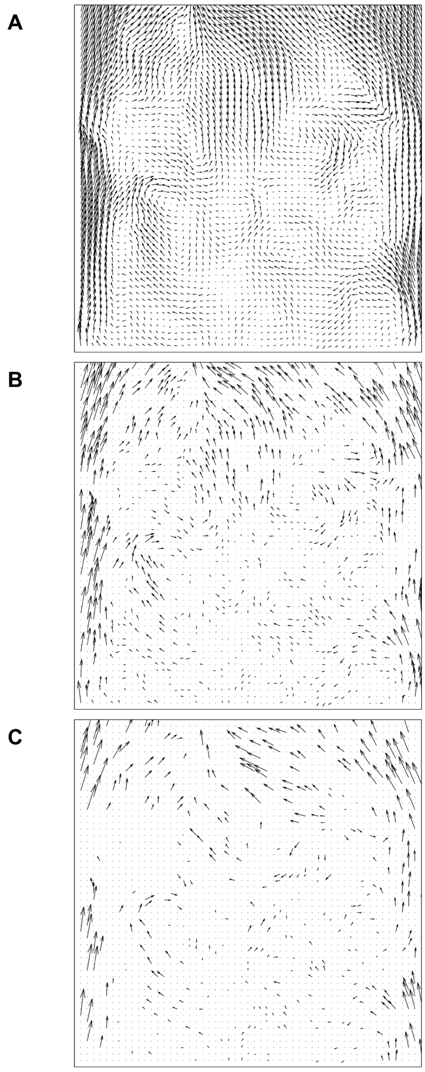

Figure (2). The horizontal cross-section of a velocity flow field modelling turbulence in the wake of a cylinder. (A) True flow, (B) with $S = 30\%$, and (C) with $S = 12.5\%$. Black dots represent empty-grid points.

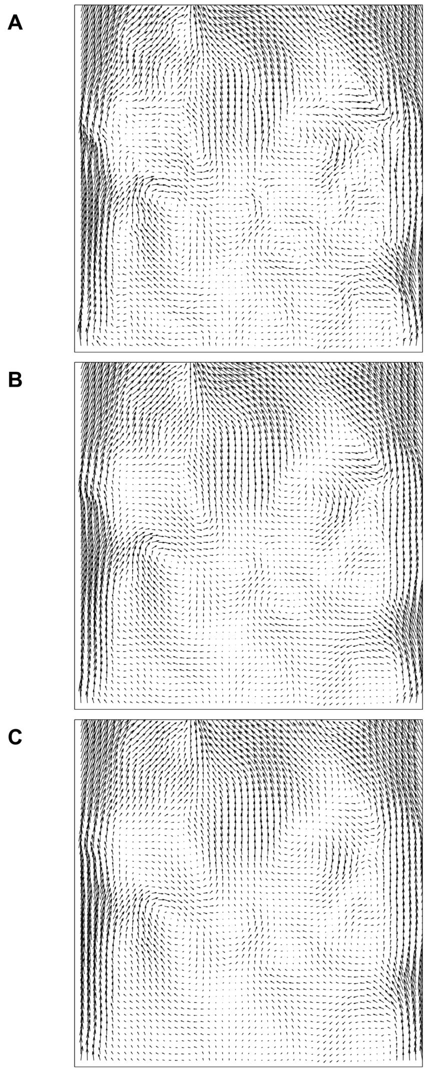

Figure (3). The horizontal cross-section of a velocity flow field modelling turbulence in the wake of a cylinder. (A) True flow, (B) PEFRA output from the restoration of Figure 2B, and (C) PEFRA output from the restoration of Figure 2C.

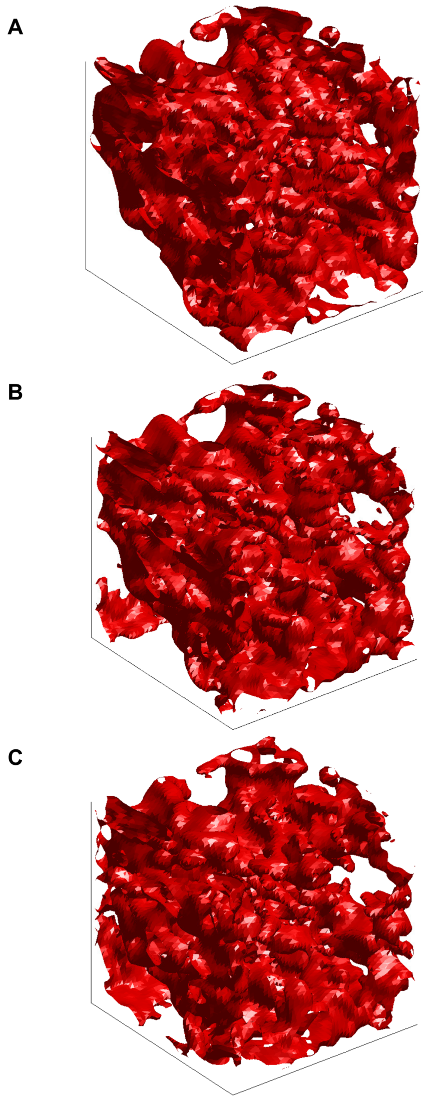

Figure (4). The three-dimensional vorticity iso-surface, corresponding to Figure 3. (A) True flow, (B) PEFRA output from the restoration of Figure 2B, and (C) PEFRA output from the restoration of Figure 2C.

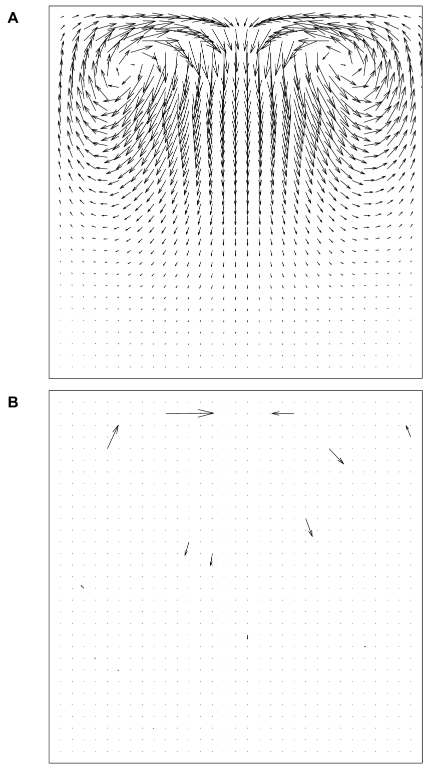

Figure (5). A vertical cross-section of the velocity flow field modelling a convection cell. (A) True flow, and (B) sparse velocity flow field where $S = 98\%$. The black dots represent empty grid-points.

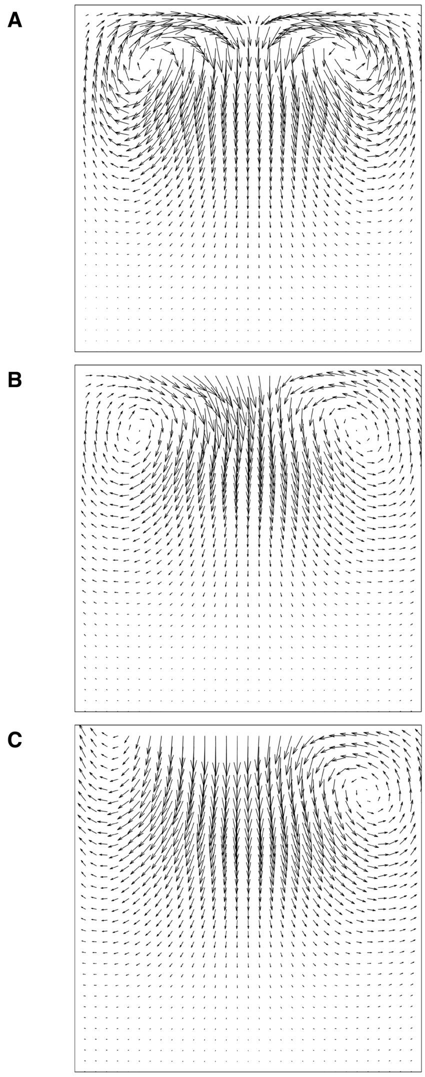

Figure (6). A vertical cross-section of the velocity flow field modelling a convection cell. (A) True flow, (B) PEFRA output from the restoration of Figure 5B. $S = 98\%$, (C) PEFRA output from the restoration of the same flow which sparsity $S = 99\%$ is below critical value ($S_{critical} = 98\%$).

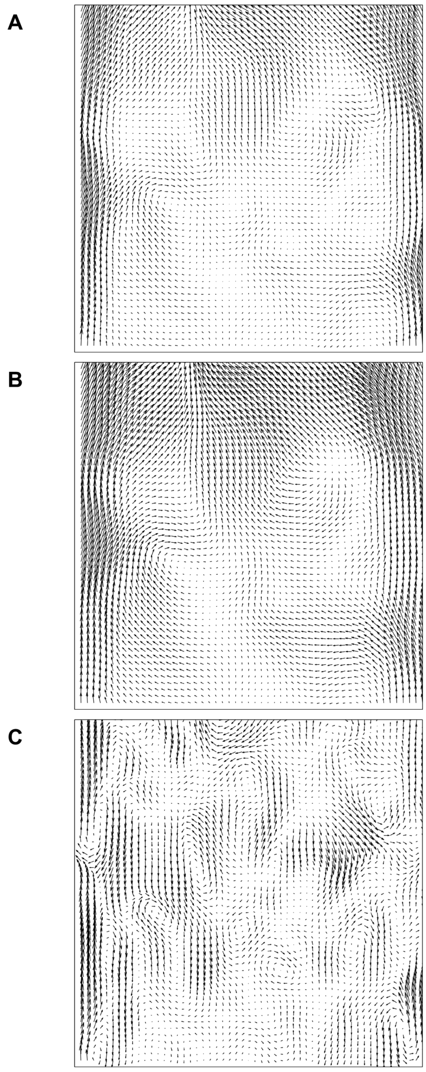

Figure (7). The horizontal cross-section of a velocity flow field modelling turbulence in the wake of a cylinder (Figure 2), reconstructed by PEFRA with (A) $\nu = 2$, (B) $\sigma = 2$ and (C)$\alpha = 3$.

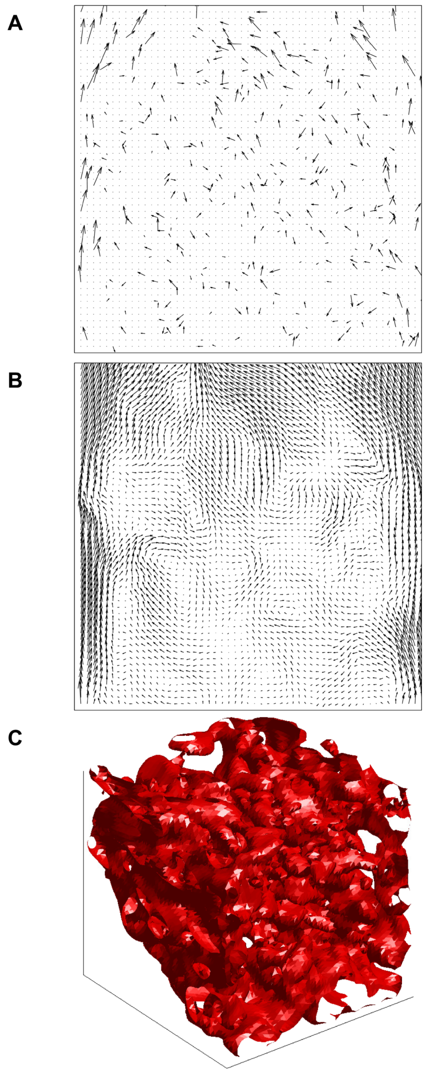

Figure (8). (A) The horizontal cross-section of a gappy and noisy velocity flow field modelling turbulence in the wake of a cylinder, and the corresponding (B) true flow and (C) vorticity iso-surface.

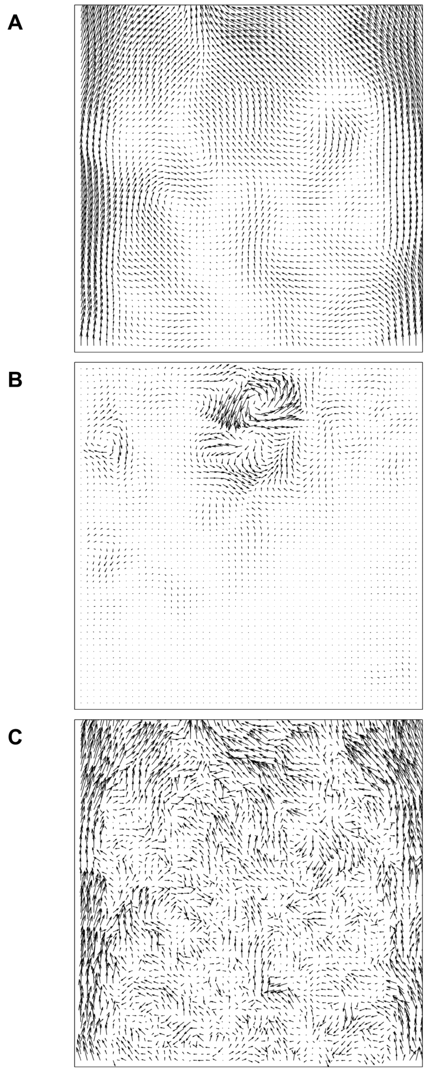

Figure (9). The horizontal cross-section of a velocity flow field modelling turbulence in the wake of a cylinder (Figure 8), reconstructed by (A) PE-FRA, (B) PCEVD and (C) AWI.

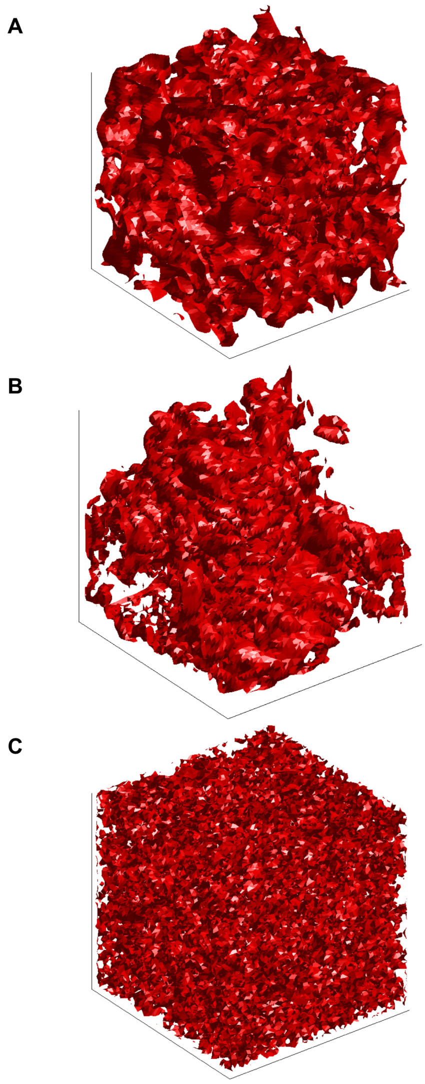

Figure (10). The three dimensional vorticity iso-surface corresponding to Figure 9, reconstructed by (A) PEFRA, (B) PCEVD and (C) AWI.

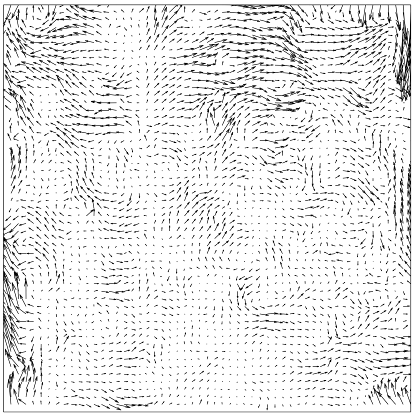

Figure (11). The difference between the true and restored field yields the vector field shown, obtained from data presented in Figure 8B and Figure 9A.

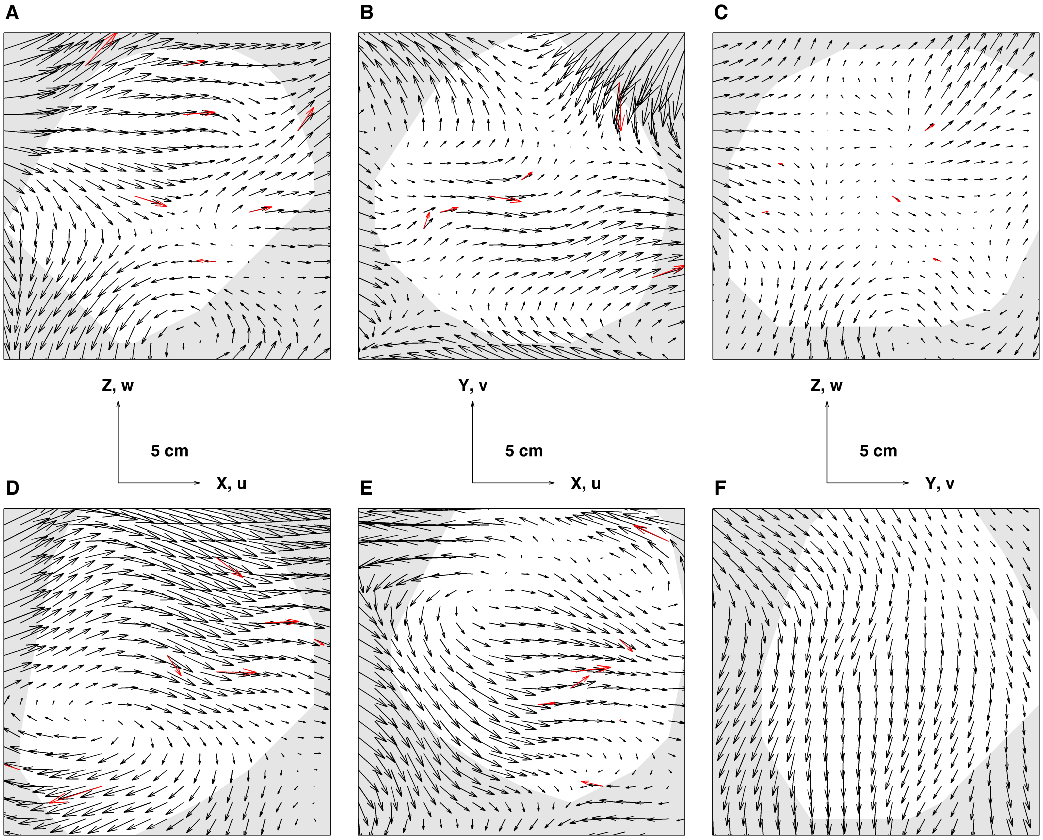

Figure (12). Row 1: cross-section of the velocity flow field corresponding to the minimum turbulence intensities recorded. Row 2: cross-section of the velocity flow field corresponding to the maximum turbulence intensities recorded. In each case, the orientation of the slices are indicated by the axes. The 3D-PTV measurements (red) and post-restoration velocity distribution (black) are overlapped. The projection of the convex hull of the tracked particles is shaded white.

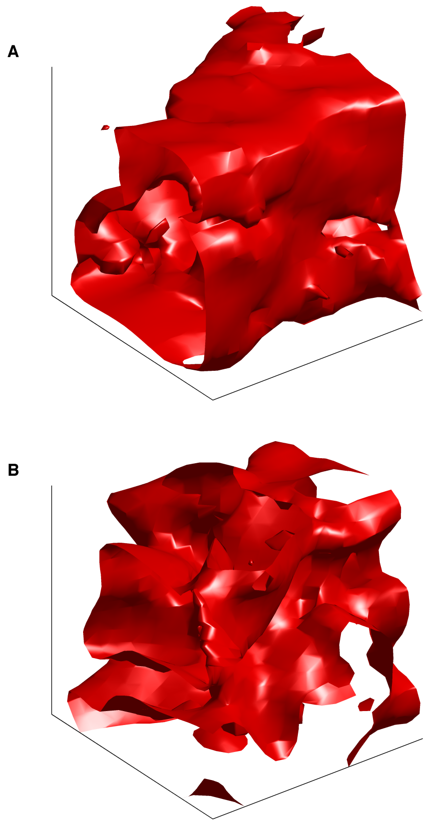

Figure (13). Vorticity iso-surfaces of the PEFRA output for the two conditions presented in Figure 12.

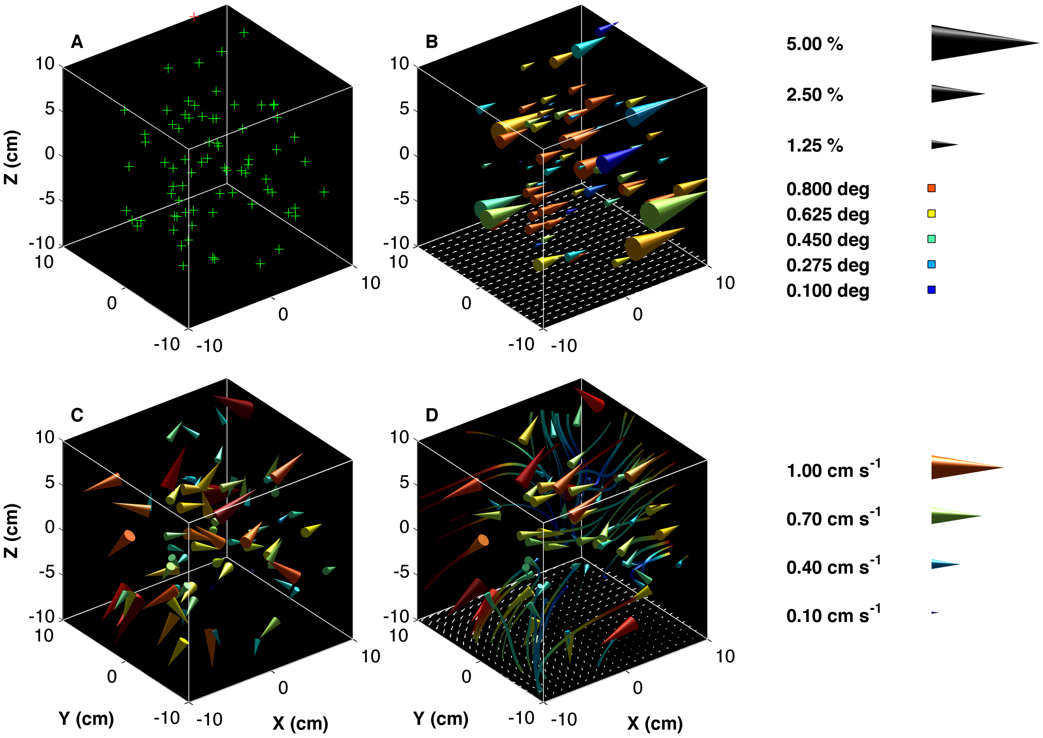

Figure (14). An instantaneous velocity flow field with a low turbulence strength: (A) output from the tracking software and gridding process; (B) The Δ_n (vector scale) and θ (vector colour) between the input and output velocity flow field at each of the seed-points; (C) Velocity distribution (coloured and scaled by the velocity magnitude) corrected by PEFRA; (D) Velocity distribution (coloured and scaled by the velocity magnitude) not corrected by PEFRA

.

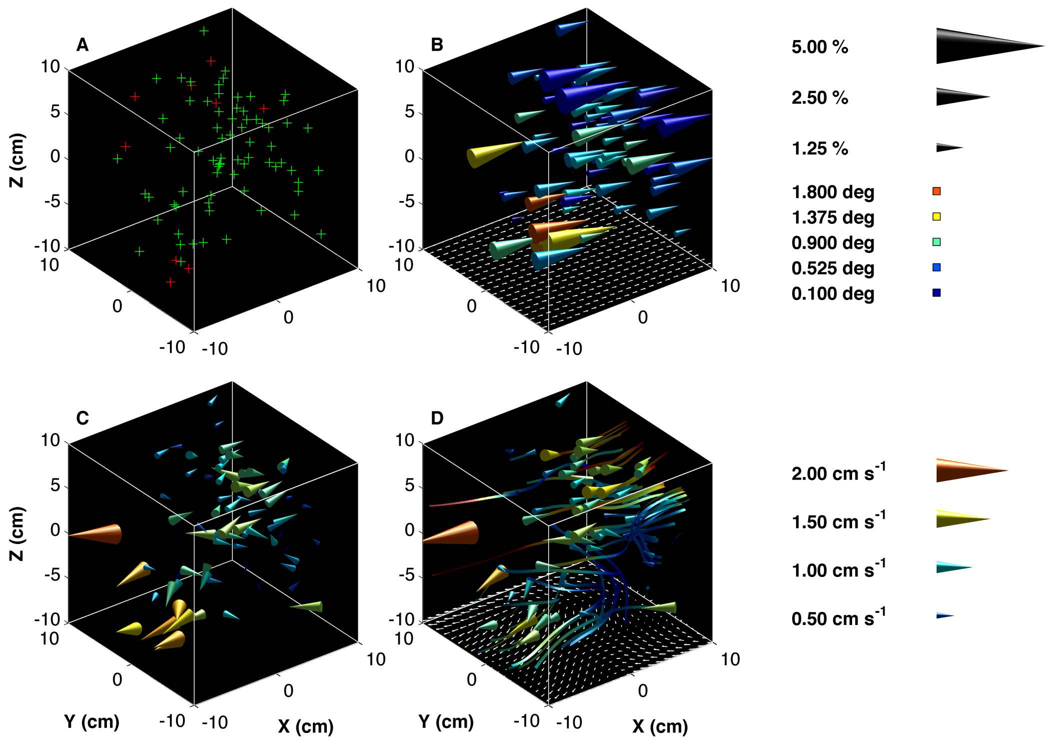

Figure (15). An instantaneous velocity flow field with a higher turbulence strength. The visualisation process is as per Figure 14.

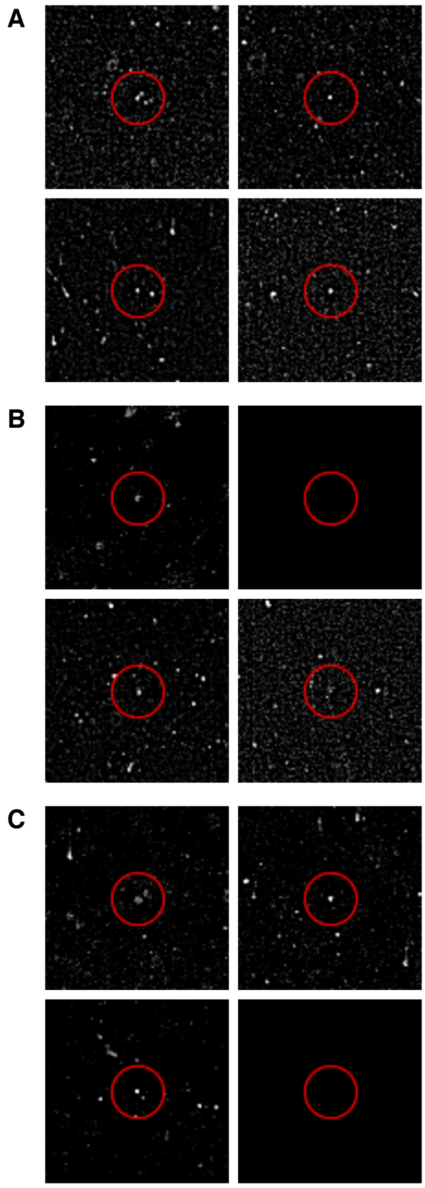

Figure (16). Three sections from the 3D-PTV image (A to C), viewed from each of the four different camera angles. The particles nearest the grid-points corresponding to: (A) the frame-minimum Δ_n; (B) the frame-minimum θ; (C) the frame-maximum Δ_n and frame-maximum θ are highlighted.

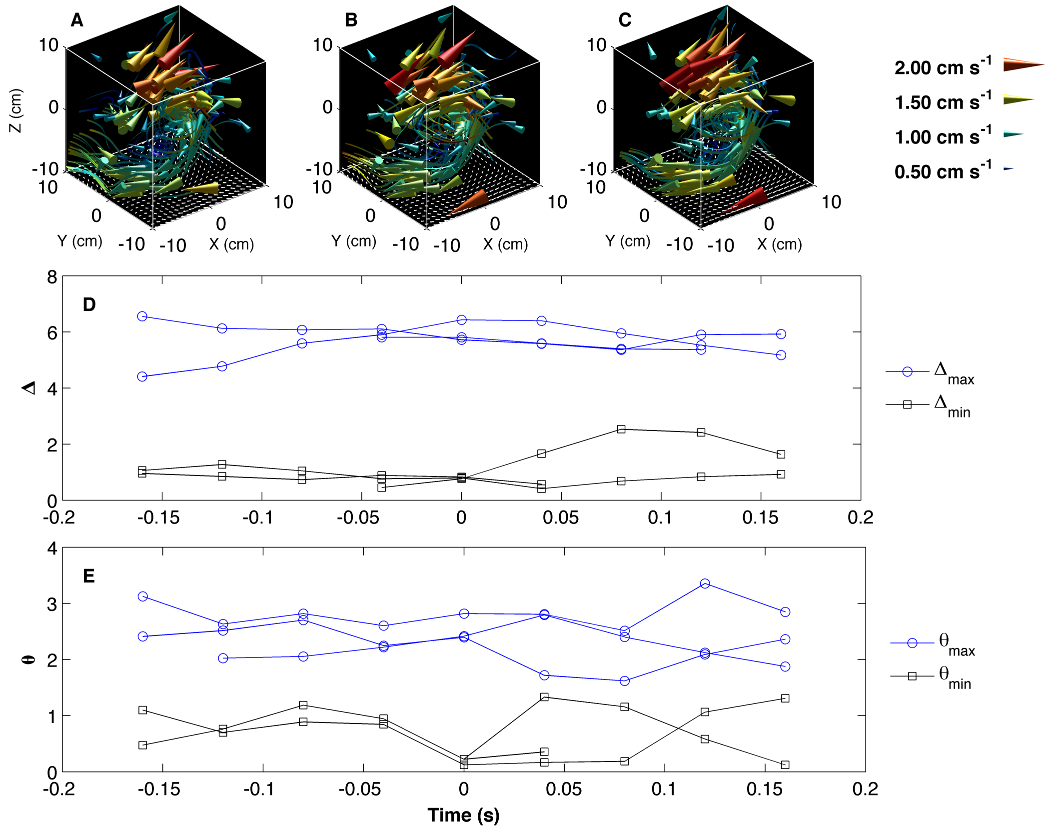

Figure (17). (A to C) Time-series of the instantaneous velocity flow field of a three-dimensional coherent structure at intervals of $1/25\,$s. Visualisation procedures are as in Figure and Figure. (D) Time-series of the adjustment made by PEFRA to 6 particles that represent the 3 maximum and 3 minimum Δ corrections made in (B) over a sequence of 7 frames. (E) Time-series of the adjustment made by PEFRA to 6 particles that represent the 3 maximum and 3 minimum θ corrections made in (B) over a sequence of 7 frames.

.

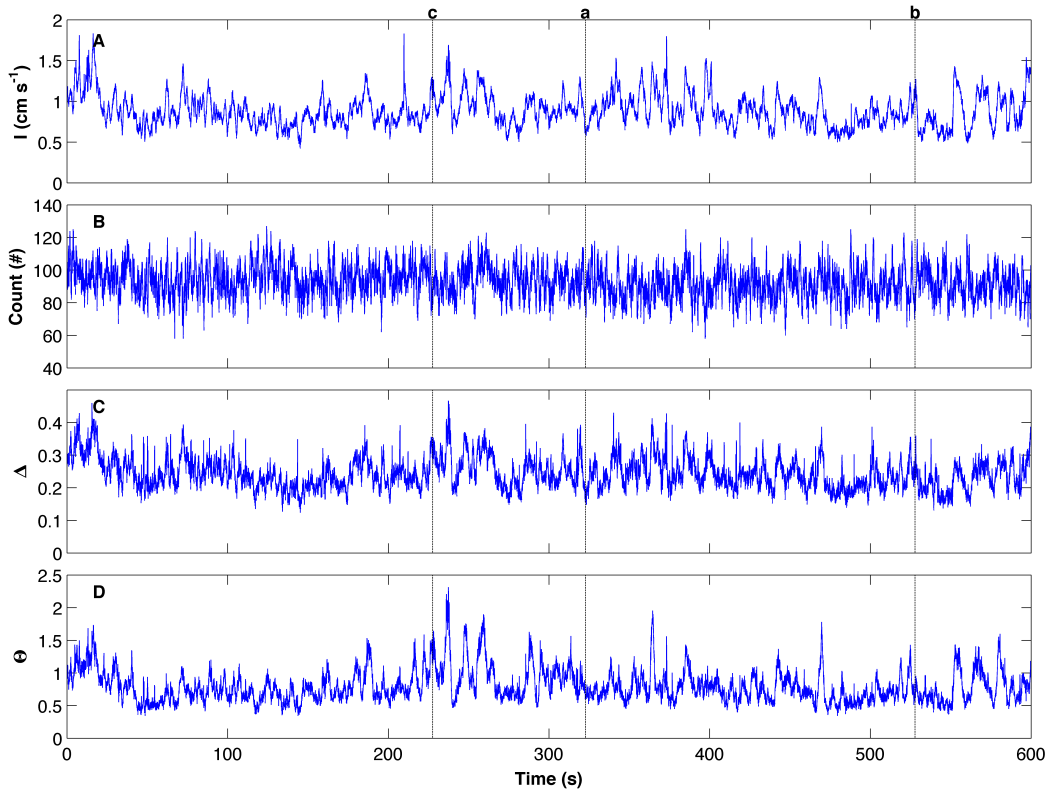

Figure (18). Time-series of the sample volume (A) mean turbulence strength, (B) total particle count, (C) frame-averaged Δ_n and (D) frame-averaged θ. The black lines represent where the velocity distributions shown in (a) Figure 14, (b) Figure 15 and (c) Figure 19 occurs in the sequence.

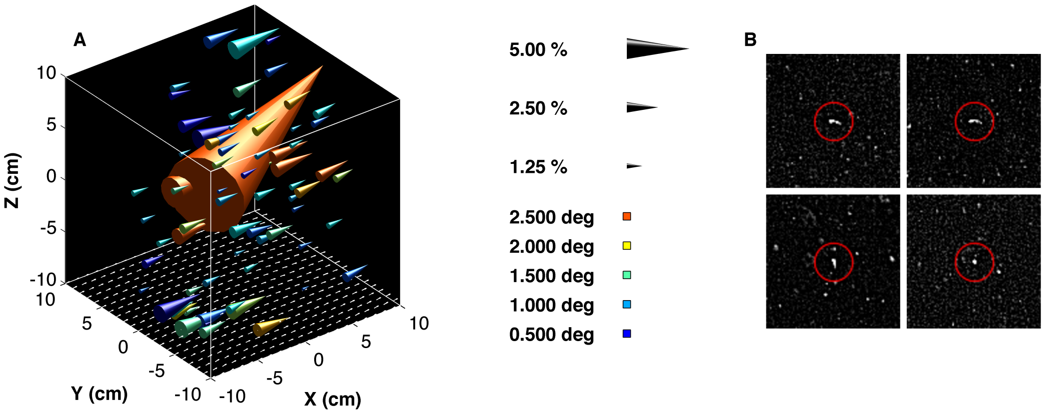

Figure (19). (A) The Δ_n and θ between the input and output velocity flow field at each of the seed-points. (B) Section from the 3D-PTV image, viewed from each of the four different camera angles, with the particle responsible for the single large vector in (A) highlighted.