

# An extended study of a generalized digital particle image velocimetry (DPIV) processing technique

J Duncan, T Bryce, H Thomsen, D Dabiri, J R Hove and M Gharib

Department of Aeronautics & Astronautics, University of Washington, Seattle, WA 98195, USA

E-mail: [dabiri@aa.washington.edu](mailto:dabiri@aa.washington.edu)

Received 7 August 2008, in final form 29 April 2009

Published 5 June 2009

Online at [stacks.iop.org/MST/20/075401](http://stacks.iop.org/MST/20/075401)

## Abstract

A generalized cross-correlation particle image velocimetry (PIV) method capable of providing direct estimates of fluid velocity, vorticity and in-plane shear rates has been introduced and described by Mayer (2000 *Exp. Fluids* **33** 443–57). While being useful, this generalized image deformation-based PIV (herein referred to as GIDPIV) method has not been deeply investigated. Toward this end, this paper examines the influence of window radii, particle diameter, seeding density, weighting functions and interpolation algorithms on the accuracy of this GIDPIV. Synthetic images representing uniform flow and an Oseen vortex flow, along with an experimental image set, are used to assess the performance results via investigation of random, bias and total errors.

**Keywords:** PIV, optimization, interpolation, weighting window, image deformation

(Some figures in this article are in colour only in the electronic version)

## 1. Introduction

### 1.1. Background

The method of particle image velocimetry (PIV) has gained much popularity in recent years as an approach to non-invasively interrogate a two-dimensional cross-section of a flow field. While having a basis in optical interference of photographic plates (Raffel *et al* 1998), modern PIV (or DPIV, digital PIV) is done almost exclusively by computer, especially with the continuous increase in computing power. Upon acquisition, the most frequently used method for post-processing images uses a cross-correlation algorithm, the most basic of which is

$$R(x, y) = \sum_{i=-K}^K \sum_{j=-L}^L I_1(i, j) I_2(i+x, j+y) \quad (1)$$

where  $K$  and  $L$  are the dimensions of the interrogation window,  $I_1$  and  $I_2$  are the intensity values of each window (time 1 and time 2) and  $R$  is the resulting correlation array. Due to the discretized nature of this method, the maximum image shift will be found at an integer pixel location. To find the true shift, a sub-pixel peak finding interpolation algorithm is

employed (e.g. bilinear, bicubic, bicubic spline, center of mass or Gaussian), where the Gaussian interpolator gives the best results (Keane and Adrian 1992).

Much work has been done toward the evolution of PIV. Westerweel *et al* (1997) showed that the displacement RMS uncertainty was  $\sim 0.002$  pixels (px) at zero displacement, increasing linearly for pixel shifts within 0–0.5 pixels, and remaining relatively flat for larger displacements. This has resulted in several window-shifting methods, where given PIV results from initially processed images, the interrogation windows were moved to the anticipated location of the particles. Westerweel *et al* (1997) showed the validity of a discrete window shift with an order-of-magnitude decrease in uncertainty. Wereley and Meinhart (2001) realized that the window shifting was acting as a forward differencing scheme ( $O(\Delta t)$  accurate) and could be improved through the use of a center difference scheme ( $O(\Delta t^2)$  accurate). Gui and Wereley (2002) furthered the window-shifting technique by interpolating the images such that they could be continuously shifted. Within such interrogation windows, this produced near-zero particle shifts, significantly reducing peak-locking effects and decreasing the bias error by nearly a factor of 5.

A common source of error to the window-shifting algorithms comes from velocity gradients within the flow, which reduce and stretch the overall correlation peak (Menieur and Leweke 2003). It was first suggested by Huang *et al* (1993a, 1993b) that the displacement should be approximated not only by translation, but rotation and dilatation as well. Local field correction (Nogueira 1999) and multi-grid methods (Scarano and Riethmuller 2000) have been developed as image distortion algorithms to account for these additional displacements. Several iterative deformation methods have been discussed by Scarano (2002). All these methods require interpolation to find the intensity values of non-integer pixel locations in the image deformation step, which leads to significantly more computational time. Further investigations into the performance of various interpolation algorithms have been performed by Astarita and Cardone (2005) and Kim and Sung (2006). The former tested the simplex, bilinear, biquadratic, bicubic, iterative discrete window offset, sinc, Fourier shift theorem and B-spline interpolators, and found the sinc, Fourier shift theorem and B-spline interpolators to be optimal. The latter tested the linear, quadratic, B-spline, Lagrange, sinc and Gaussian interpolators and found that both sinc and Lagrangian interpolators were optimal, despite increases in computational time over quadratic or B-spline interpolators. The B-spline interpolators tested by Kim and Sung were of lower order than those tested by Astarita and Cardone, which is perhaps why they were not found to be optimal. A more recent study performed by Astarita (2008) investigated the effect of the interpolation algorithm choice in the dense predictor step of deformation methods, that is, the step before deforming the image. Of the interpolators tested (bilinear, shifted bilinear, B-spline and ideal), it was found that a B-spline interpolator was most accurate, with the second order being adequate. The shifted bilinear interpolation scheme was also found to give good results but it introduced noise at lower frequencies.

A problem noted when using a continuous window-shifting or deformation-based algorithm is the lack of convergence of certain interrogation windows' displacement upon subsequent iterations. Nogueira (1999) found that this was due to the moving-average behavior of the cross-correlation function, which, when the spatial frequency of the image produced a negative response, resulted in the errors not approaching zero despite multiple iterations. Nogueira (1999) found that if the proper weighting function

$$w_{pp}(\xi, \eta) = 9 \left( 4 \left| \frac{\xi}{F} \right|^2 - 4 \left| \frac{\xi}{F} \right| + 1 \right) \times \left( 4 \left| \frac{\eta}{F} \right|^2 - 4 \left| \frac{\eta}{F} \right| + 1 \right) \quad (2)$$

was applied to each window before cross-correlation, the frequency response could be made to never become negative. This improvement allows higher resolution when used in a local field correction manner (LFC) wherein structures smaller than the interrogation window can be correctly identified. A further investigation into the choice of weighting windows was done by Astarita (2007), introducing more classical weighting windows such as Blackman, Gauss, Harris, Kaiser–Bessel,

Nuttal, etc. In addition, the use of two weighting windows was investigated, finding that the use of a top-hat window in the so-called weighted average step allows the stabilization of the algorithm for most weighting windows. Nogueira's window was found to perform well for most applications.

### 1.2. Generalization of the PIV cross-correlation technique subject to affine transformations

Alternative to iterative approaches, it has been proposed that the image-matching process can be considered an optimization problem using the cross-correlation as the function to maximize (Mayer 2002). In this manner, shear, rotation, expansion and translation are used as parameters in the optimization and are thus directly known. By performing a first-order Taylor expansion of the velocity field and breaking it into its translation, rotation and linear deformation components, the image shift can be written as

$$S_{u,v,\omega,s_{ij}} = \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} s_{11} & s_{12} - \omega \\ s_{12} + \omega & s_{22} \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} \quad (3)$$

where  $u$  and  $v$  are the translational velocity components,  $\omega$  is the rotation,  $s_{ij}$  is the deviatoric strain tensor and  $x_c$  and  $y_c$  are the coordinates of the interrogation window center point. Because the shifting operation in equation (3) is not a discrete process, it is applied to the continuous coordinates  $x$  and  $y$ , not the discrete coordinates  $i$  and  $j$ . Therefore, Mayer interpolated the image intensity fields at  $(x, y)$  from its  $(i, j)$  coordinates, resulting in

$$I_1(x, y) = Q(I_1(i, j)) \quad (4)$$

$$I_2(x, y) = Q(I_2(i, j)) \quad (5)$$

where  $I_1 = I_1(i, j)$  and  $I_2 = I_2(i, j)$  are the two-dimensional matrices describing the intensity fields, with  $Q$  being an appropriate interpolation algorithm. For convenience purposes, Mayer collectively stored the velocity components  $u, v, \omega$  and  $s_{ij}$  in a six-dimensional matrix  $q$ :

$$q = (u, v, \omega, s_{11}, s_{22}, s_{12}). \quad (6)$$

Although the operation in equation (3) describes the shift that the particles at  $t_1$  undergo to align with the particles at  $t_2$ , in implementation, Mayer constructed a virtual interrogation area at a new time,  $t_h = \frac{t_1+t_2}{2}$ , centered halfway between  $t_1$  and  $t_2$ . Therefore, the particles at  $t_1$  and  $t_2$  can be thought of as shifts of the virtual interrogation window at  $t_h$ .

To implement this generalized procedure, Mayer used a circular virtual interrogation window, with discrete data points spaced, on average, about 1 pixel apart. This spacing gave the total number of data points per interrogation window as

$$N = \pi R^2, \quad (7)$$

where  $R$  is the radius of the virtual interrogation window. The two shifting operators in equations (4) and (5) applied to this circular interrogation window at  $t_h$  result in the new particle locations. To calculate the image intensity at the data points in the interrogation windows at  $t_1$  and  $t_2$ , Mayer used a standard bilinear interpolation algorithm.

Because the deformed interrogation windows at  $t_1$  and  $t_2$  have a discrete number of data points, the weighted normalized correlation function must be written in its discrete form. This form is given by Mayer in equation (8):

$$C(I_1, I_2) = \frac{\sum_i \{[w(r_i)]^2 I_1(x_i, y_i) I_2(x_i, y_i)\}}{\sqrt{\sum_i \{w(r_i)[I_1(x_i, y_i)]^2\} \sum_i \{w(r_i)[I_2(x_i, y_i)]^2\}}}, \quad (8)$$

where the summation is performed over all data points. To ensure that the function  $C$  in equation (8) is a smooth function of  $q$ , Mayer chose to use the following weighting function:

$$w(r) = 1 - \left(\frac{r}{R}\right)^2. \quad (9)$$

Unlike the two-dimensional correlation matrices observed in standard DPIV, the correlation function in equation (8) must be optimized for all six parameters of  $q$ . To accomplish this process, Mayer used an optimization routine given by Powell (1982). Using this optimization approach, it was found that both random and bias errors decrease with increasing particle image diameters with no minimum up to the maximum tested diameter of 6 pixels. Also, the bias error follows the same trend of peak locking as for standard DPIV using a discrete window shift. Lastly, Mayer also found that the bias error is largest close to the center of the vortex where the velocity gradients are highest.

While Mayer's generalized method and results are very good, the parameters tested have been limited to only bilinear interpolation, inverse parabolic weighting function and two particle diameters. In the present study, we extend Mayer's study to determine the effects of various interpolation routines, particle diameters, seeding densities, weighting functions, radii and noise on the bias and random errors of this generalized method. Section 2 presents the algorithm procedure, interpolation routines and weighting functions used; section 3 presents the simulated flow images and their associated parameters; section 4 presents the results and section 5 discusses these results; and section 6 presents the conclusions.

## 2. Overview of the algorithm procedure and parameters

### 2.1. Algorithm procedure

The first step in the algorithm consists of identifying information that will be used to process the images. First, a user-defined border is applied to the images specifying the regions not used in any calculations. This border is used to ensure that the virtual interrogation window at  $t_h = \frac{t_1+t_2}{2}$  (centered halfway between the two times  $t_1$  and  $t_2$ ) is never deformed off of the image. Next, the interrogation window center points are specified. This defines the grid points at which data (i.e. velocity, vorticity, strain rates) will be assigned. Once this grid has been determined, the number and location of data points within each interrogation window are calculated based on the specified radius of that window. The value of a chosen weighting function is then calculated

for each data point based on either radial or  $x$  and  $y$  distance from the center of the window.

After such information is determined, the second step is to generate a starting estimate for the six values of the  $q$  matrix (see equation (6)), where the entries in  $q$  are defined by the transformation operator (see equation (3)). These estimates are obtained by first processing the images with a window-shifting cross-correlation and outlier detection algorithms (i.e. Westerweel *et al* 1997) to obtain the  $u$  and  $v$  velocities. The resulting  $u$  and  $v$  velocities are then used to determine the  $\omega$ ,  $s_{11}$ ,  $s_{22}$  and  $s_{12}$  components of the  $q$  matrix (see e.g. Raffel *et al* 1998).

The final step in the GIDPIV algorithm is to find  $q$  that best estimates the deformations within the processed images. This process includes deforming the interrogation window, interpolating the image intensity, performing the cross-correlation and finding the optimum deformation using the optimization routine given in Powell (1982). In order to continuously deform the interrogation window, an interpolation algorithm must be chosen beforehand. As with most optimization algorithms, Powell is designed to find a function minimum; so the argument of Powell is changed to  $1 - C$ , where  $C$  is the correlation value given in equation (8).

### 2.2. Interpolation algorithms

The three interpolation algorithms presently studied are bilinear, bicubic and bicubic spline. Bilinear interpolation is the most commonly used interpolation algorithm because of its computational speed. Bicubic and bicubic spline interpolation algorithms are more accurate than bilinear interpolation but take significantly more computational time as well. The actual code to perform the bilinear, bicubic and bicubic-spline interpolations has been taken from Press *et al* (2002).

### 2.3. Weighting functions

It is clear from Nogueira (1999) that under certain circumstances, a weighting function can increase the accuracy of a given PIV algorithm. Toward this end, seven weighting functions are studied and are described as follows: constant, parabolic down (not normalized), parabolic down (normalized), parabolic down (shifted up), parabolic up, parabolic up (shifted up) and Nogueira's weighting function. Aside from the weighting function labeled not normalized, they all are chosen so that the area underneath them is equal to 1. These weighting functions are given in table 1. Other weighting functions have been proposed in the literature, but Nogueira's weighting function has continually returned good performance (Astarita 2007), and will be considered as the most advanced for the current work.

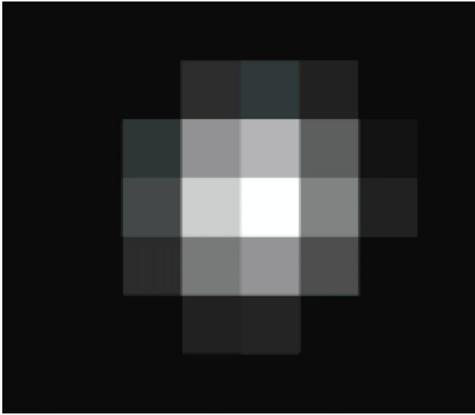
## 3. Flow simulation and error description

### 3.1. Image generation

The generated images used to test this algorithm are 8 bit gray-scale images with the intensity at each pixel ranging from 0 (black) to 255 (white). The particles can be modeled

**Table 1.** Summary of weighting functions and corresponding equations.

Weighting function name	Weighting function equation
Constant	$w_c(r) = \frac{1}{\pi R^2}$
Parabolic down (not normalized)	$w_{pd}(r) = 1 - \left(\frac{r}{R}\right)^2$
Parabolic down	$w_{pdn}(r) = \left(\frac{2}{\pi R^2}\right) \left(1 - \left(\frac{r}{R}\right)^2\right)$
Parabolic down (shifted up)	$w_{pdsu}(r) = \left(\frac{2}{3\pi R^2}\right) \left(2 - \left(\frac{r}{R}\right)^2\right)$
Parabolic up	$w_{pu}(r) = \left(\frac{2}{\pi R^4}\right) r^2$
Parabolic up (shifted up)	$w_{pusu}(r) = \left(\frac{2}{3\pi R^2}\right) \left(1 + \left(\frac{r}{R}\right)^2\right)$
Nogueira's	$w_{pp}(\xi, \eta) = 9 \left(4 \left \frac{\xi}{F}\right ^2 - 4 \left \frac{\xi}{F}\right  + 1\right) \left(4 \left \frac{\eta}{F}\right ^2 - 4 \left \frac{\eta}{F}\right  + 1\right)$

**Figure 1.** Sample computer-generated particle with center intensity  $I_0 = 255$  and  $d_p = 4.0$ .

as Gaussian, following Raffel *et al* (1998). However, a computer-generated particle must be represented with discrete pixels, so the intensity value at each pixel location in the particle is determined by integrating over the area of the pixel. The actual length of the particle diameter is defined as the distance between the intensities one standard deviation from the maximum on the Gaussian distribution. An example of a computer-generated particle is shown in figure 1.

The last step in generating synthetic images is filling the laser sheet volume with the generated particles. This process is done by first specifying a required particle density (also referred to as seeding density) for the first image. In this case, the particle density is defined as the density of the visible particles projected onto the  $x, y$  plane. A small volume is then filled with the correct number of particles, each with a random value for its center point,  $x_0, y_0, z_0$ . If any particles overlap, each pixel location is given only the higher of the competing intensities. This process is then repeated until the entire first image is generated. The second image is then generated by displacing each particle according to a predetermined velocity profile. For the second image, particles are allowed to be displaced outside of the small sections in which they are generated or off the image entirely. The synthetic image pairs used in this study included no noise, did not incorporate out-of-plane losses and used a box light sheet. An example of three generated images with different seeding densities is shown in figure 2.

For the uniform flow case, the  $x$ -direction displacement is varied from 0.0 to 1.5 px in steps of 0.1 px (the  $y$ -direction displacement is zero). Seeding densities of 1%, 2% and 5% 1/px are used. The particle diameter ranged from 1.3 to 4.9 px in steps of 0.4 px. Interrogation window radii of 4, 8, 16 and 32 px were tested.

For the Oseen vortex flow case, the particles are displaced according to

$$u_\theta = \frac{\Gamma}{2\pi r} \left[1 - e^{\left(\frac{-r^2}{\gamma}\right)}\right], \quad (10)$$

where  $u_\theta$  is the vortex velocity in the  $\theta$  direction,  $r$  is the distance from the center of the vortex and  $\Gamma$  and  $\gamma$  are parameters determining the strength of the vortex. From this velocity profile, the  $z$ -direction vorticity can be computed using equation (11):

$$\omega_z = \frac{\Gamma}{\gamma\pi} \left[e^{\left(\frac{-r^2}{\gamma}\right)}\right]. \quad (11)$$

For this study,  $\Gamma = 2400\pi \frac{\text{px}^2}{\text{s}}$  and  $\gamma = 5000 \text{ px}^2$  giving a maximum velocity gradient of approximately  $0.23 \frac{1}{\text{px}}$ . The resulting velocity and vorticity profiles are shown in figure 3.

In this case, all images are  $512 \times 512$  px, with seeding densities of 2% and 5% 1/px, and particle diameters of 1.3, 2.5, 3.7 and 4.9 px. All of the cases tested are summarized in table 2.

### 3.2. Error description

In general, two different types of error can be used to describe the accuracy of a particular measurement, the bias error and the random error. The bias error describes the difference in a calculated averaged value from the exact value. It exists in all measurements as a deviation from the true value and is described as

$$\varepsilon_{\text{bias}} = u_m - u_{\text{exact}}, \quad (12)$$

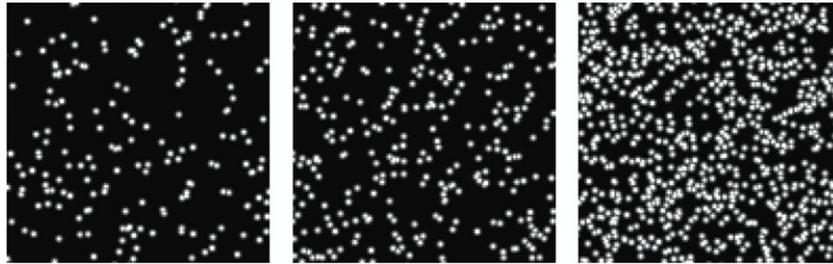
where  $u_{\text{exact}}$  is the single true value and  $u_m$  is the average measured value given as

$$u_m = \frac{1}{N} \sum_{i=1}^N u_i, \quad (13)$$

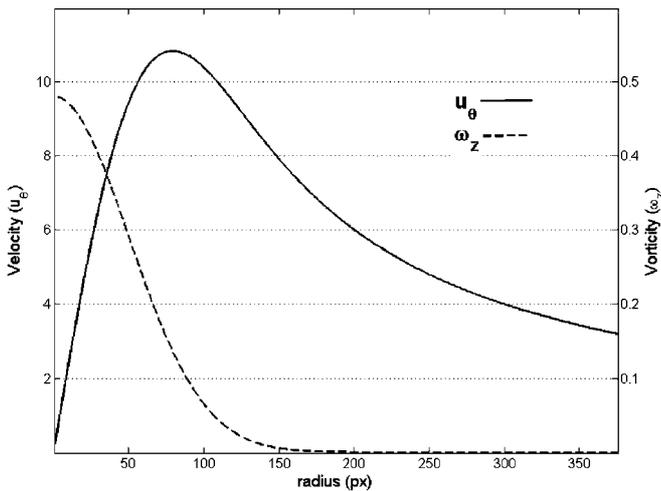
with  $u_i$  being a single measured value. The random error describes the statistical spread of the measured values about the

**Table 2.** Cases tested using the GIDPIV algorithm.

Parameter	Uniform flow	Oseen vortex
Image size (px)	124 × 124, 184 × 184, 304 × 304, 544 × 544	512 × 512
Light sheet	Box	Box
Border (px)	32	32
x, y distance between center points (px)	Interrogation window radius	Interrogation window radius
Radius (px)	4, 8, 16, 32	4, 8, 16
Seeding density, $N_i$ (1/px)	1%, 2%, 5%	2%, 5%
Particle diameter, $d_p$ (px)	1.3–4.9 (0.4 increments)	1.3, 2.5, 3.7, 4.9
Weighting function	a, b, c, d, e, f, g	a, c, f, g
Interpolation	Bilinear, bicubic, bicubic spline	Bilinear, bicubic, bicubic spline



**Figure 2.** Computer-generated images with seeding densities  $N_i = 1\% \text{ 1/px}$  (left),  $N_i = 2\% \text{ 1/px}$  (middle) and  $N_i = 5\% \text{ 1/px}$  (right).



**Figure 3.** Specific Oseen vortex velocity and vorticity profiles used in the present research.

average measured value. It is the result of random inaccuracies in the measurement process, and is expressed as

$$\epsilon_{\text{random}} = \left( \frac{1}{N} \sum_{i=1}^N (u_i - u_m)^2 \right)^{\frac{1}{2}} \quad (14)$$

The total error is a combination of both the bias error and the random error and is calculated by taking the root sum squared of the two:

$$\epsilon_{\text{total}} = \sqrt{\epsilon_{\text{bias}}^2 + \epsilon_{\text{random}}^2} \quad (15)$$

**3.2.1. Uniform flow.** As described above, to accurately represent the error of the present GIDPIV algorithm, the calculations must be averaged over many repetitions. For

the uniform flow cases, only one image pair is required per displacement/variable set because each interrogation window will have the same displacement. Therefore, the calculated velocity of each interrogation window can be averaged with that of all the other interrogation windows of the same image pair leading to the following descriptions for uniform flow:

$$u_{m\_UF} = \frac{1}{N_{cp}} \sum_{i=1}^{N_{cp}} u_i \quad (16)$$

$$\epsilon_{\text{random\_UF}} = \left( \frac{1}{N_{cp}} \sum_{i=1}^{N_{cp}} (u_i - u_{m\_cp})^2 \right)^{\frac{1}{2}} \quad (17)$$

where  $N_{cp}$  is the number of interrogation window center points in the images. For the present study, a total of 196 interrogation windows are used for each image pair. With this requirement, the size of the generated images then depended on the interrogation window radius.

**3.2.2. Oseen vortex.** For the Oseen vortex cases, the velocity and vorticity at each interrogation window center point vary with distance from the vortex center, so the only calculations that can be averaged together are those that are at the same radial location. Thus, multiple image pairs must be evaluated for each variable set. Accordingly, the following equations result for the Oseen vortex cases:

$$u_{m\_OV} = \frac{1}{N_R * N_{IP}} \sum_{i=1}^{N_R * N_{IP}} u_i \quad (18)$$

$$\epsilon_{\text{random\_OV}} = \left( \frac{1}{N_R * N_{IP}} \sum_{i=1}^{N_R * N_{IP}} (u_i - u_{m\_OV})^2 \right)^{\frac{1}{2}} \quad (19)$$

where  $N_R$  is the number of points at the same radial location and  $N_{IP}$  is the number of image pairs. For this study, 100 image pairs are evaluated for each variable set to generate the above-mentioned errors. Finally, in each of the above-mentioned error equations,  $u$  can represent any calculated quantity. For example, the above equations are used to calculate the error in the  $x$ -direction velocity for the uniform flow cases and the error in both total velocity,  $U = \sqrt{u^2 + v^2}$ , and vorticity for the Oseen vortex cases.

## 4. Results

### 4.1. Uniform flow results

For the sake of brevity, the results of the investigation of the use of GIDPIV on a uniform flow field will be given rather than the data. Several properties were altered in order to determine the combination resulting in the highest possible accuracy. These properties were interrogation window radius, particle image diameter, particle seeding density, window weighting function and interpolation algorithm. The approach taken was to start with commonly accepted property values (Raffel *et al* 1998), such as a constant weighting function, a seeding density of 5% 1/px, bilinear interpolation and a particle diameter of 3.7 px. With these properties, the optimal window size was found. Using the optimal window size, the particle image diameter was adjusted to find the optimum. This process was repeated with all the properties until the optimum was reached.

Window radii of 4, 8, 16 and 32 pixels were tested with the larger windows showing significantly lower random, and thus also total, error. The 16 px and 32 px windows showed approximately the same total error. The random error showed a peak-locking aspect with a 1 pixel periodicity, similar to that found by Gui and Wereley (2002). The lowest maximum total error was produced by the 32 px radius case, at 0.025 px. Different parameters (seeding density, particle diameter, etc) can produce even lower errors, but all follow approximately the same trend, with larger windows producing lower errors.

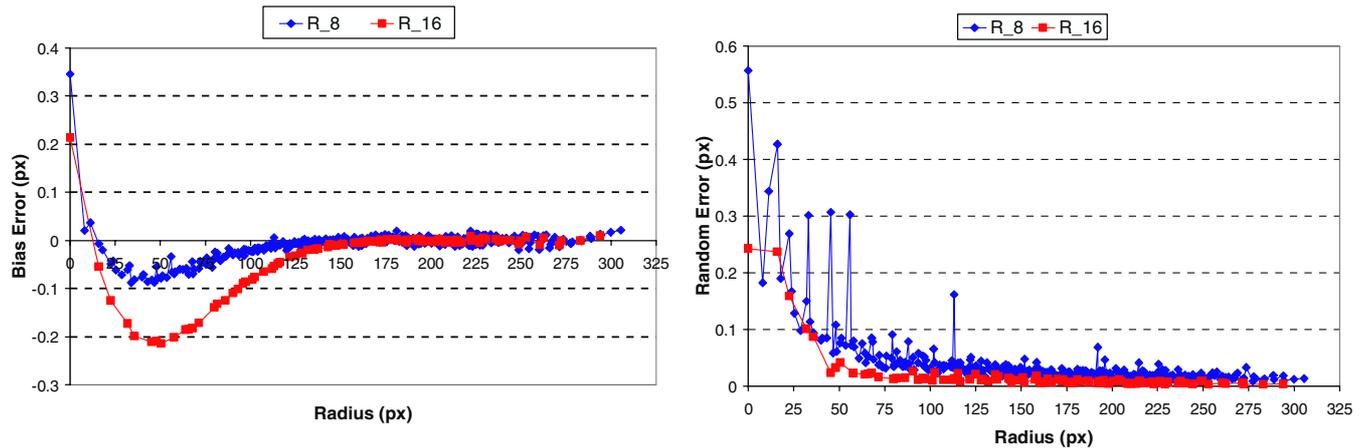
With the interrogation window size fixed at 32 px, the particle image diameter was varied from 1.3 to 4.9 px in steps of 0.4 px. For both bias and random errors, it was found that increasing diameter resulted in lower error, with the lowest total error at 0.018 px. Mayer (2002) also found similar behavior, showing that increasing particle diameters (up to 4 px were tested) resulted in lower errors. These results are not consistent with traditional DPIV (e.g. Raffel *et al* 1998), which finds an optimal particle image diameter of around 2.1–2.5 px. This trend of decreasing error with increasing particle diameter is also found for all parameter sets as a function of the particle diameter. While even larger diameters may be optimal, 4.9 px was the highest tested in this work resulting in a maximum total error of 0.018 px.

Using a 32 px radius window and particles with a diameter of 4.9 px, the seeding density was varied from 1 to 5% in order to determine an optimum. The seeding density of  $N_i = 1\%$  1/px had the lowest bias error and the highest random error. The highest seeding density,  $N_i = 5\%$  1/px, on the other hand, had the highest bias error and the lowest random

error. Peak locking was also evident for both the bias and random errors, although it decreased with increasing particle image diameter similar to what was noticed by Kim and Sung (2006). However, because the random error is much lower in magnitude than the bias error, the bias error dominates the total error. The seeding density of 2% shows a slightly larger total error than the 1% seeding density errors. Therefore, with all other variables held constant and a large interrogation window radius, the seeding density of 1% actually gives the lowest total error at 0.012 px. This behavior is different from that seen for standard DPIV (Gui and Wereley 2002). For larger windows, the trend of lower error with decreasing seeding density is found for all parameter sets. Smaller windows benefitted from higher seeding densities, presumably due to the relatively low amount of information contained in the windows.

Using the weighting functions defined in section 2.3, it was found that weighting functions did not have a very large effect on the bias error. This behavior makes sense because the uniform flow is the same everywhere in the interrogation window. Therefore, increasing or decreasing the importance of a certain region should not affect the bias. The random error shows the Nogueira and parabolic up weighting functions to have the highest random error. This is most likely due to the severely diminished amount of data within the windows for these functions. The Nogueira and parabolic up weighting functions remove most of the data from the window. The parabolic down (shifted up) and constant weighting functions result in the lowest random (and therefore total) errors. This is most likely due to the fact that no area of the window is exaggerated or diminished relative to the rest. In a uniform flow, this approach results in the lowest errors. Nogueira *et al* (1999) also noticed a new source of error when using the Nogueira weighting function termed the slippage error. It was advised that this weighting function was not to be used on windows smaller than 32 by 32 px (Nogueira *et al* 2001). Although the current window has a diameter of 64 px, this could be an indication as to why Nogueira's weighting function results in significantly higher errors with respect to other weighting functions. These trends hold for all parameter sets as well, where both the constant and parabolic down (shifted up) weighting functions consistently give good results. Since the constant weighting function was used above, no decrease in maximum total error was found.

Since the GIDPIV algorithm requires the image to be deformed with subpixel accuracy, some sort of interpolation algorithm is required. Three commonly used interpolators are the bilinear, the bicubic and the bicubic spline algorithms. Here, these interpolators were tested using parameters found to be optimal in the previous paragraphs. Both the bias and random errors, and thus the total error, were drastically affected by the choice of interpolation algorithm. This was especially apparent in the case of bias error, which was decreased by nearly an order of magnitude by the use of bicubic spline over bilinear interpolator. The bicubic interpolator also showed a significant improvement over the bilinear, but only by about a factor of 2. The random error showed similar results, with bicubic offering a 25% error reduction and bicubic spline offering a 40% reduction over the bilinear interpolator. For



**Figure 4.** Velocity bias (left) and random (right) error as a function of the radial distance from the center of the Oseen vortex for two radii.  $N_i = 2\%$   $1/\text{px}$ ,  $d_p = 4.9$  px, constant, bilinear interpolation.

the bilinear interpolator, the bias and random error were comparable such that the total error received contributions from both. In the case of bicubic and bicubic spline, the random error dominated the total error. All of these error reductions are simply due to the higher accuracy of the more advanced interpolation algorithms. These improvements do come at a cost in terms of computational time. For one data set, using bilinear interpolation resulted in a computational time of about 1 min. For the same data set, the bicubic and bicubic spline interpolators took 30 min and 2.5–3 h, respectively. Other investigations (Astarita and Cardone 2005, Kim and Sung 2006) into the use of more advanced interpolation algorithms have yielded similar results with bicubic giving better results than bilinear, and bicubic spline resulting in even lower errors. For example, Astarita and Cardone found that bicubic interpolation reduced a random error by more than a factor of 2 over bilinear, and bicubic spline decreased it by another factor of 2. Using the optimal parameters found above, a maximum total error of only 0.006 px is possible for uniform flow.

#### 4.2. Oseen vortex results

**4.2.1. Window radius comparison.** For the Oseen vortex case, three interrogation window radii are studied, 4, 8 and 16 px. The errors are prohibitively large for the 4 px case (on the order of 0.5 px for all radial distances), and thus are discarded from the current discussion. The 8 and 16 px radius windows are tested for 100 sample images and the averaged results are shown in figure 4. A seeding density of 2% is used along with a particle diameter of 4.9 px in accordance with the results of the uniform flow testing.

The error is clearly higher in the center of the vortex where the gradients are greatest. An interrogation window of 8 px radius gives a lower error near the center of the vortex, although with significantly more scatter. The 16 px radius window results in a lower total error at larger distances. The 4 px radius is just too small to accurately capture the large velocity gradients of the Oseen vortex. On the other hand,

the 16 px radius is large enough that it does not experience a large loss or addition of particles, so its random error is very low. However, because it is so large, the velocity changes too much within the interrogation window, and the assumption that velocity is constant within the window is no longer very accurate. Therefore, its bias error is higher than that of the 8 px radius. The 8 px radius seems to be the best compromise between the two and, as a result, gives a low total error in the regions of higher velocity gradients.

Figure 5 shows the total error in vorticity for 8 and 16 px radius windows. The same parameters as above are used.

In this case, interrogation windows with a 16 px radius have the lowest error for all radial distances. Beyond about 75 px, the error essentially goes to zero, but below 75 px, it appears that even a 16 px radius is too small to accurately reflect the rotation of the flow. The 16 px window is the superior choice; it exhibits lower scatter in high gradients, produces the lowest error at large radii (low gradients) and reflects rotation with the least error (radii greater than 75 px have a negligible total error in vorticity). For these reasons, 16 px windows are used for the remainder of the analysis.

**4.2.2. Particle diameter comparison.** The results from the uniform flow case reveal that larger particle diameters result in lower errors. This is assumed to be true for the Oseen vortex case as well, but is tested here. The seeding density is held at 2%, again in accordance with the results of the uniform flow studies. An interrogation window radius of 16 px is used due to its performance in terms of vorticity error. Figure 6 shows the results obtained by varying the particle diameter from 1.3 to 4.9 px.

Again, the largest particle diameter is shown to result in the lowest error, particularly in the case of a random error. This result was also found by Mayer (2002), up to a diameter of 4 px (largest tested). The decrease in error is most evident in the center where velocity gradients are the highest. Again, this behavior differs from traditional DPIV (e.g. Raffel *et al* 1998) where particles with a diameter between 2.1 and 2.5 px produce the lowest errors. Figure 7 shows

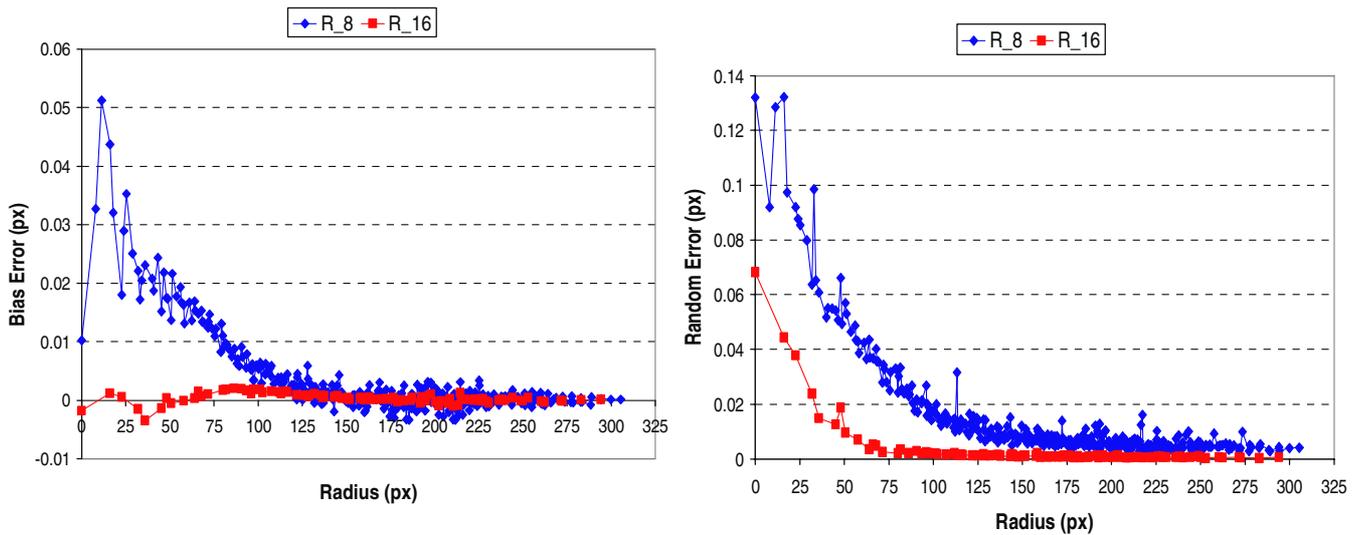


Figure 5. Vorticity bias (left) and random (right) error as a function of the radial distance from the center of the Oseen vortex for two radii.  $N_i = 2\% 1/\text{px}$ ,  $d_p = 4.9 \text{ px}$ , constant, bilinear interpolation.

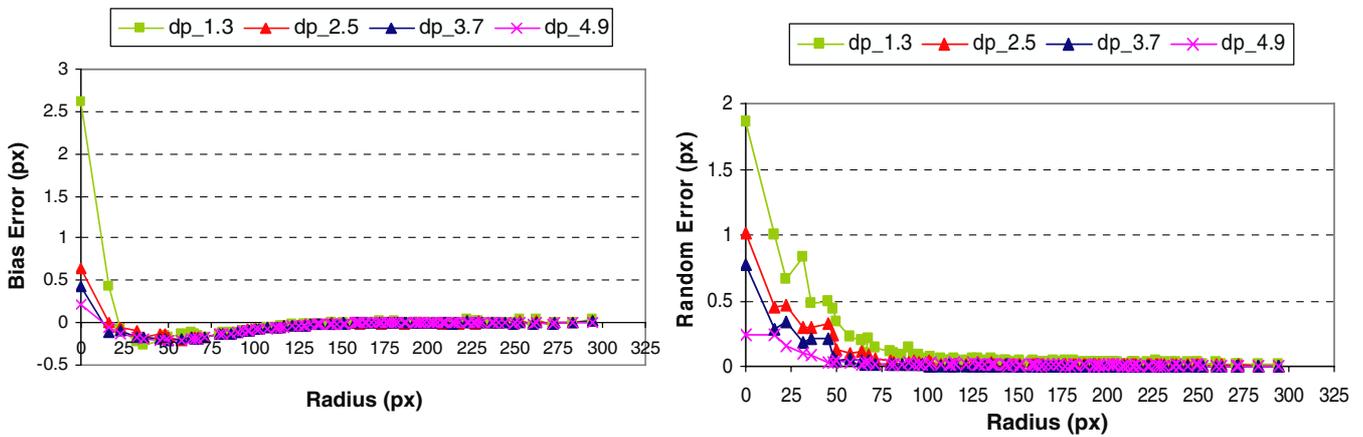


Figure 6. Velocity bias (left) and random (right) error as a function of the radial distance from the center of the vortex for various particle diameters.  $R = 16 \text{ px}$ ,  $N_i = 2\% 1/\text{px}$ , constant, bilinear interpolation.

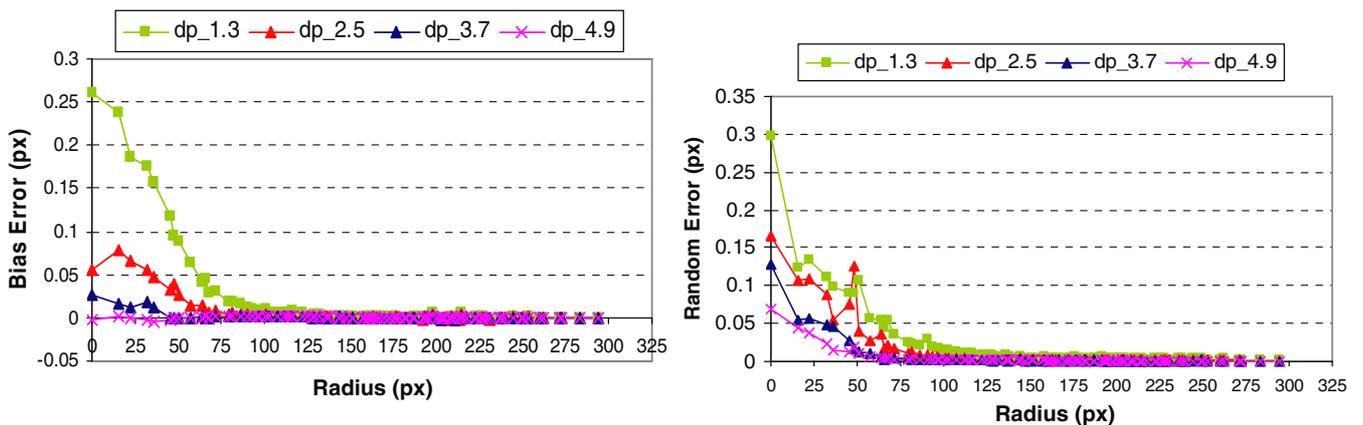
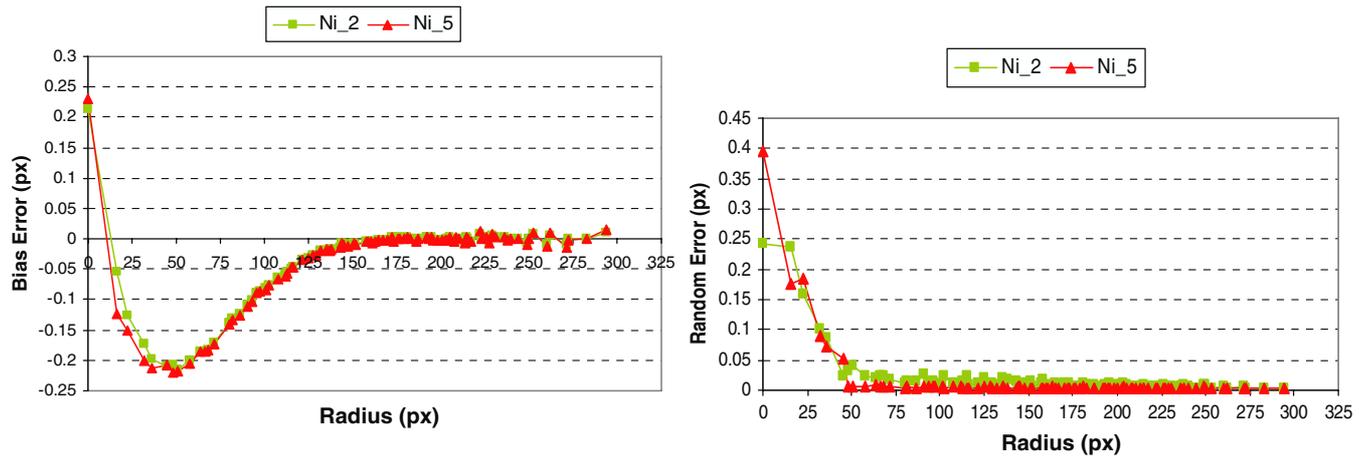


Figure 7. Vorticity bias (left) and random (right) error as a function of the radial distance from the center of the vortex for various particle diameters.  $R = 16 \text{ px}$ ,  $N_i = 2\% 1/\text{px}$ , constant, bilinear interpolation.

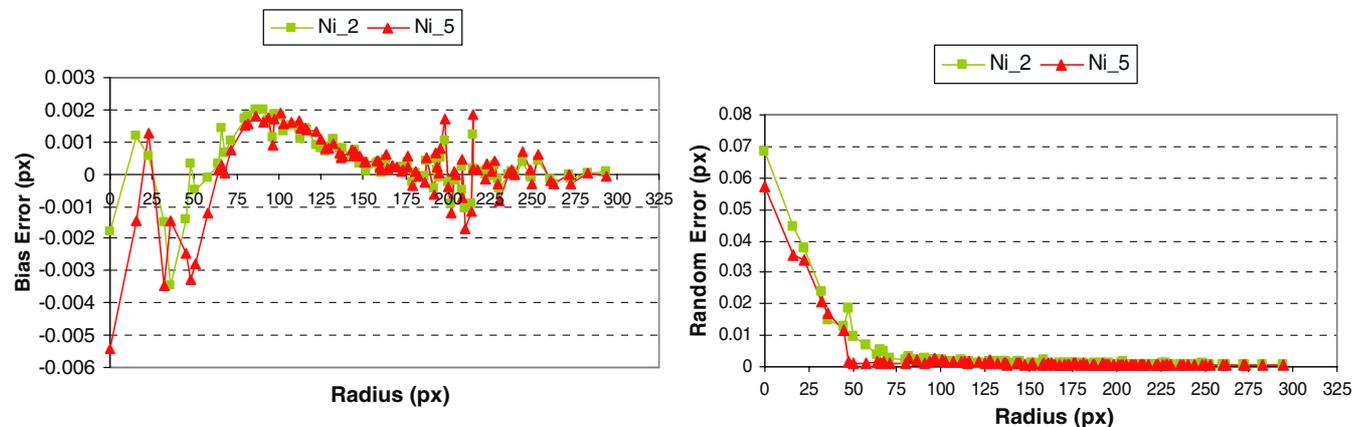
the effect of particle image diameter on the error in vorticity measurements.

The trend of decreasing error with increasing particle image diameter is even more pronounced for vorticity

measurements than velocity. In this case, however, both bias and random errors show a decrease. As in the case of uniform flow, it cannot be said that a diameter of 4.9 px results in the lowest error as diameters greater than this were not tested.



**Figure 8.** Velocity bias (left) and random (right) error as a function of the radial distance from the center of the Oseen vortex for two seeding densities.  $R = 16$  px,  $d_p = 4.9$  px, constant, bilinear interpolation.



**Figure 9.** Vorticity bias (left) and random (right) error as a function of the radial distance from the center of the Oseen vortex for two seeding densities.  $R = 16$  px,  $d_p = 4.9$  px, constant, bilinear interpolation.

**4.2.3. Seeding density comparison.** Using the same parameters as before, the effect on error in both velocity and vorticity is tested. The interrogation window radius is 16 px, the particle image diameter is 4.9 px and a constant weighting function and bilinear interpolation are used. Figure 8 shows the effect of seeding density on the bias, random and total error of velocity.

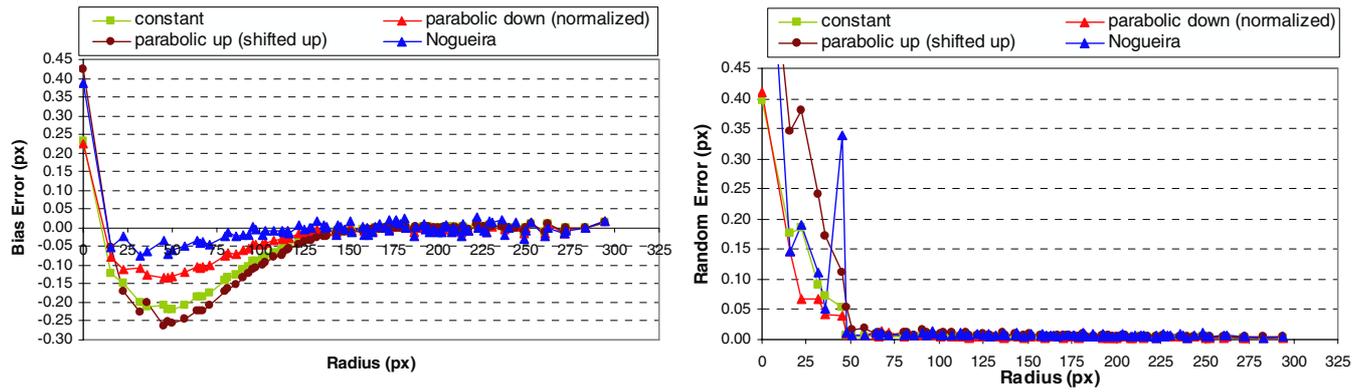
The effect of seeding density is not nearly as pronounced for the Oseen vortex case as it is in the uniform flow case, but at distances greater than 125 px from the vortex center, the higher seeding density exhibits about half the error of the lower seeding density. This appears to come mainly from the random error contribution, although the exact reason is unknown. It could be that more information is contained in the interrogation window. The uniform flow case was investigated using an interrogation window with a radius of 32 px, meaning that it had four times the area of the current window, making the added information from the higher seeding density unnecessary. Figure 9 shows the effect of seeding density for error in vorticity measurements.

For vorticity, the higher seeding density gives a slightly lower error than the lower seeding density. The lower error is most clearly visible in the random error, close to the vortex

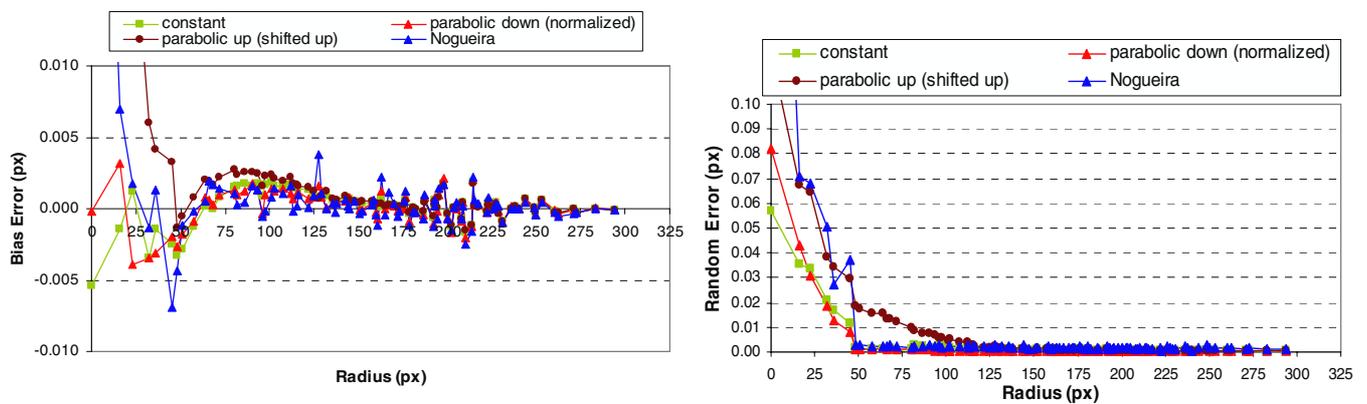
center. Although not shown, interrogation windows with a radius of 8 px (four times less area than the current window) benefit even more from an increased seeding density, again indicating that more information in the window results in the lower error, particularly the random error.

**4.2.4. Weighting function comparison.** Of the weighting functions described in section 2.3, only four showed promise in the case of the Oseen vortex; these are the constant, the parabolic up (shifted up), the parabolic down (normalized) and Nogueira’s weighting function. The effect of these weighting functions on bias and random errors for both velocity and vorticity is investigated. Figure 10 shows the results for velocity.

Nogueira’s weighting function gives the lowest total error up to 125 px from the vortex center. However, it has the highest scatter of the three at lower radii as indicated in the random error plot. Additionally, at higher radii, Nogueira’s weighting function results in slightly higher error than the other three, which are all approximately equal. Nogueira’s weighting function performs best in areas of high gradient, followed by the parabolic down weighting function. This



**Figure 10.** Velocity bias (left) and random (right) error as a function of the radial distance from the center of the Oseen vortex for various weight functions.  $R = 16$  px,  $N_i = 5\%$  1/px,  $d_p = 4.9$  px, bilinear interpolation.



**Figure 11.** Vorticity bias (left) and random (right) error as a function of the radial distance from the center of the Oseen vortex for various weight functions.  $R = 16$  px,  $N_i = 5\%$  1/px,  $d_p = 4.9$  px, bilinear interpolation.

is due to the focus of both functions on the center of the window, reducing the effect of velocity gradients on the edges. Figure 11 shows the effects of weighting function choice on error in the measurement of vorticity.

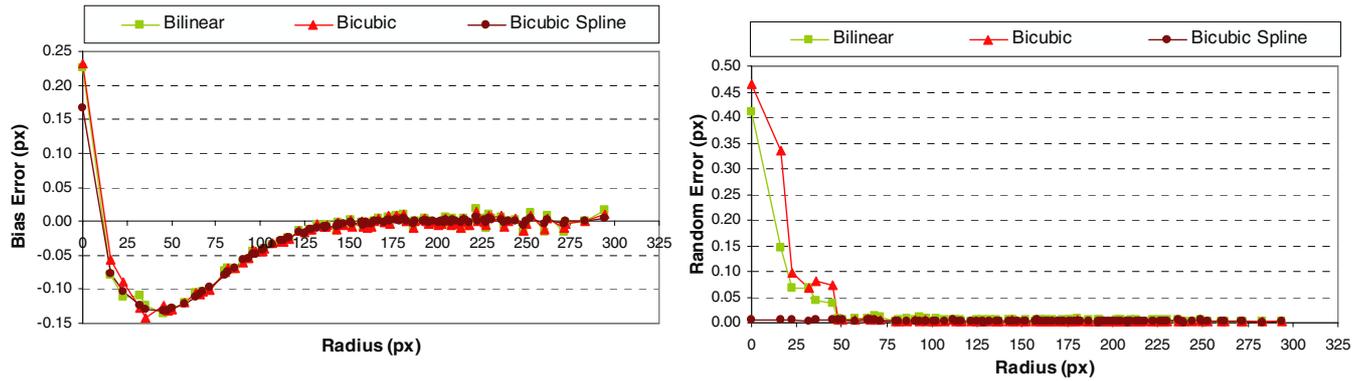
In the case of vorticity, parabolic down and constant weighting functions perform the best, with constant providing the lowest error near the vortex center (mainly a benefit in random error), but parabolic down providing a lower error further from the center (mainly in the bias error). Since it was found earlier that a larger interrogation window performs better near the vortex core, it is intuitive that the constant weighting function gives lower error than parabolic down, as the latter emphasizes the center of the window over the edges. In overall performance, between velocity and vorticity measurements, the parabolic down (normalized) weighting function seems to be a good compromise with Nogueira’s weighting function also performing well.

**4.2.5. Interpolation algorithm.** Since the current technique relies on image deformation, the choice of interpolation algorithm obviously should have a large effect on the accuracy of the technique. The same three common interpolation algorithms tested for the uniform flow case are investigated here. The other parameters have been chosen based on the results of previous sections. The interrogation window has a

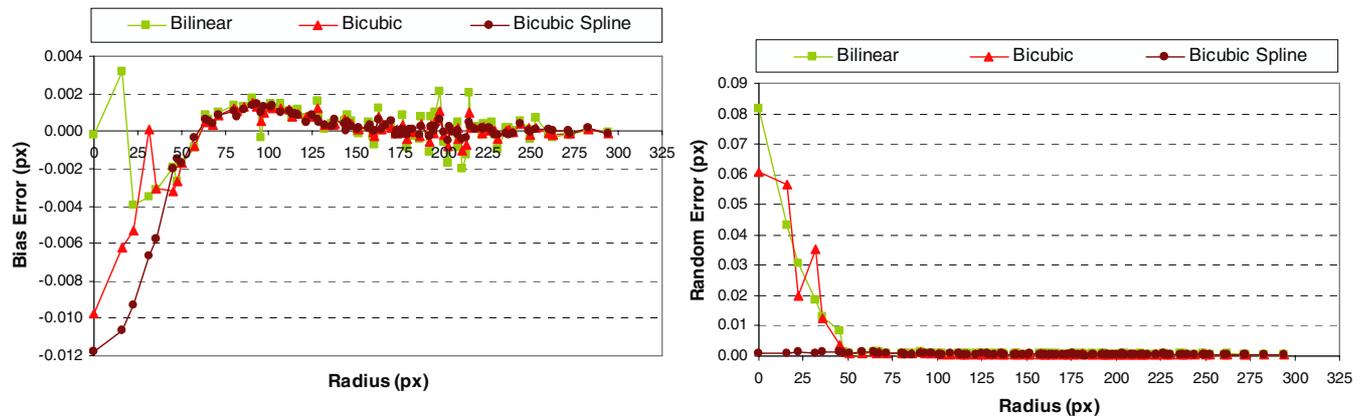
radius of 16 px, the particle image diameter is 4.9 px, the seeding density is 5% and a parabolic down (normalized) weighting function has been used. Figure 12 shows the effect of the interpolation algorithm choice on the bias and random errors in the measurement of velocity.

The choice of interpolation algorithm shows remarkably little effect on the bias error, but can significantly decrease the random error. However, since the total error is dominated by the bias error, the more accurate interpolators (bicubic and bicubic spline) do not show much improvement beyond 50 px from the vortex center. Near the vortex center, the bilinear and bicubic interpolators perform approximately equivalent, with the bicubic spline offering a dramatic improvement, with the random error no greater than 0.015 px. Figure 13 shows the effect of interpolator choice on vorticity measurement.

Here, the bilinear interpolator performs best in terms of bias error, where it has its lowest error up to about 50 px from the vortex center. For the random error, the bicubic interpolator did not show an improvement over the bilinear interpolator, but the bicubic spline interpolator decreased the random error drastically. Near the vortex center, the bicubic spline reduces the total error to slightly more than 0.01 px, and at radii greater than 50, the error decreases to  $\sim 0.003$  px. Since the random error dominates the total error, the bicubic spline interpolator is clearly superior in spite of having the higher bias error.



**Figure 12.** Velocity bias (left) and random (right) error as a function of the radial distance from the center of the Oseen vortex for various interpolation algorithms.  $R = 16$  px,  $N_i = 5\%$  1/px,  $d_p = 4.9$  px, parabolic down (normalized).



**Figure 13.** Vorticity bias (left) and random (right) error as a function of the radial distance from the center of the Oseen vortex for various interpolation algorithms.  $R = 16$  px,  $N_i = 5\%$  1/px,  $d_p = 4.9$  px, parabolic down (normalized).

For the error in both velocity and vorticity, the bicubic interpolator did not result in improvements over the bilinear interpolator, which is much faster computationally. The bicubic spline interpolator showed drastic improvement in the random error for both velocity and vorticity, which is to be expected as it is a significantly more advanced interpolation algorithm.

#### 4.3. Comparison with standard PIV

Since the computational cost of the GIDPIV algorithm is quite large, a comparison with standard (Raffel *et al* 1998) PIV has been performed to show the clear benefit of the more advanced processing. The final case of the Oseen vortex was run with a non-window-shifting cross-correlation algorithm, the results of which were checked for outliers using a standard normalized median technique. Figures 14 and 15 show a comparison of the results from this standard PIV with the results of GIDPIV. The seeding density was 5% and the particle image diameter was 4.9 px. The standard PIV used  $32 \times 32$  px windows with a 50% overlap, while the GIDPIV algorithm used the same parameters as in section 4.2.5.

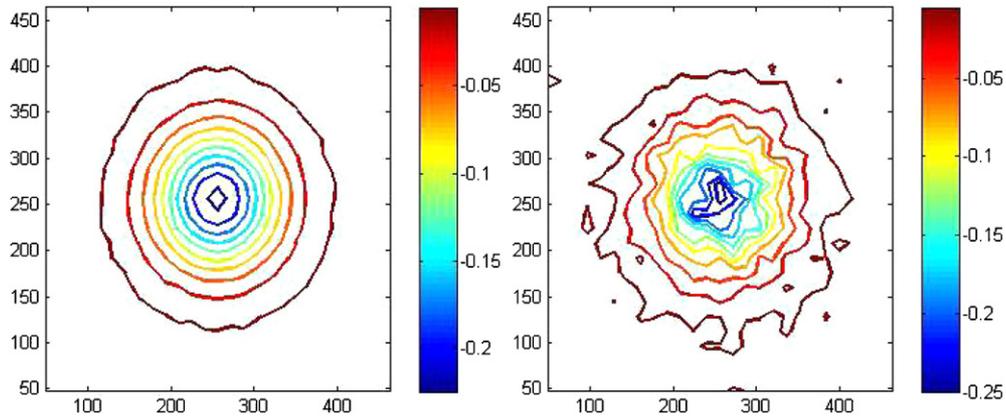
It is clear that the GIDPIV algorithm more accurately measures the velocity profile of the Oseen vortex, and more impressively, accurately calculates the vorticity directly. Even

though standard PIV has a smoothing effect due to its similarity to a moving average, the vorticity contours of the GIDPIV algorithm are much smoother, accurately representing the Oseen vortex. While the processing time may be significantly longer, the benefits to accuracy are worth the cost.

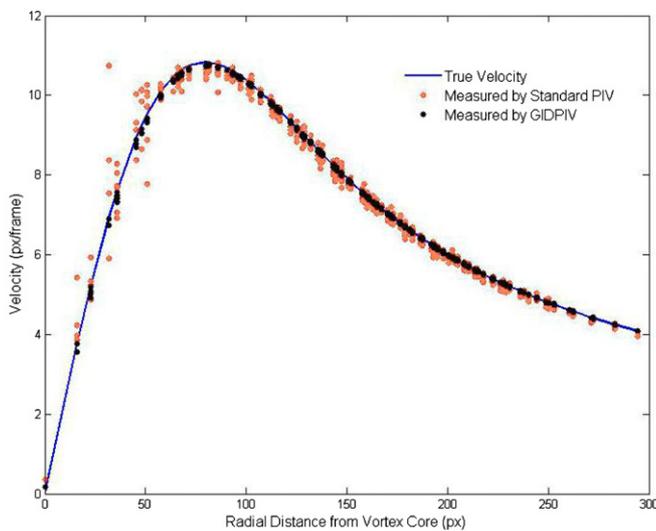
#### 4.4. Experimental image

Since the purpose of the GIDPIV algorithm is to experimentally determine flow properties, it is necessary to test it on real images, which contain noise, out-of-plane particle loss and other factors making PIV more difficult. A shear layer image set was chosen as a test due to the various flows visible along with some three-dimensionality present in certain regions of the flow. The area viewed was 22 cm by 22 cm, with flow velocities of  $10.5 \text{ cm s}^{-1}$  and  $22.5 \text{ cm s}^{-1}$  and a Reynolds number based on visual shear thickness of  $1.2 \times 10^4$ . The apparatus and details of the flow in question are described by Dabiri (2003). Figure 16 shows a sample image from the set.

While the image is fairly clean, there is still some unavoidable error and also, the average particle image diameter is significantly lower (about 2 px) than what was found to be optimal. In addition, many of the particle images are clipped to an intensity value of 255, indicating a slight overexposure. The



**Figure 14.** Comparison of the vorticity contours from GIDPIV (left) and standard PIV (right) from the Oseen vortex of section 4.2.



**Figure 15.** A plot of the velocity magnitude as a function of the radial distance shows that the GIDPIV results have far less scatter and more accurately measure the true velocity profile as compared to standard PIV.

user-controlled parameters used for this image set were chosen based on the results of the two previous sections. A 16 px radius was attempted with Nogueira's weighting function and a bicubic spline interpolation. However, the results showed a significant slippage error so a window radius of 32 px was used instead. The results of the GIDPIV processing are shown in figure 17.

It is clear that the current algorithm is able to pick up structures within the flow quite well and that it is not adversely affected by the loss of particles or image noise. There is still potential for improvement if the particle size were increased in the experiment such that the particle image diameter is around 5 px.

## 5. Discussion

### 5.1. Uniform flow

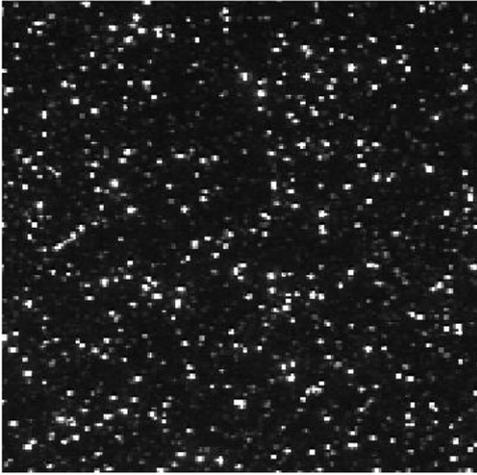
For the uniform flow cases, the total error almost always decreases with increasing interrogation window radius. The

reason for this trend is that the larger interrogation windows cover more area of the flow, and, since the flow is uniform, larger windows have more particle pairs to contribute to the correlation. Because the smaller interrogation windows cover less flow area, they are more susceptible to errors from particles entering and leaving them. Larger particle image diameters give lower overall errors, with no maximum size found in the current study. This result holds for all parameter sets, and is one major difference of GIDPIV as compared to conventional DPIV. Another fundamental difference is the lower error as a result of lower seeding density; however, this only holds for larger interrogation windows, as smaller windows require more information in the form of higher seeding density to remain accurate. The weighting function is found to have little effect on bias error, given that the flow is uniform. The random error is quite susceptible to the choice of weighting function with the Nogueira and parabolic up functions giving the highest error, while constant and parabolic down (shifted up) functions the lowest error. One of the most evident reductions in error comes from the choice of interpolation algorithm. As can be expected, bilinear performs worst, with bicubic giving a 40% reduction in total error and bicubic spline a further 25% reduction. The choice of interpolator may be influenced by the increased computational time, especially for bicubic spline.

Overall, the best results were found for an interrogation window radius of 16 or 32 px, a seeding density of 1% (given a large window), a particle diameter of 4.9 px, a constant or parabolic down (shifted up) weighting function and a bicubic spline interpolator. With this parameter set, maximum total errors as low as 0.006 px are possible.

### 5.2. Oseen vortex

The velocity total error is lowest for the  $R = 8$  px cases near the center of the vortex (at a radius of less than about 50 px) and the  $R = 16$  px cases further away from the center of the vortex. On the other hand, the vorticity total error is lowest for the  $R = 16$  px cases for the entire range of radial distances. Unlike the uniform flow cases, the larger seeding density gives the best results for both the velocity and the vorticity calculations. However, the total error still decreases with increasing particle



**Figure 16.** 100 by 100 px sample image from the experimental shear layer set.

diameter. The choice of interpolation algorithm affects mainly the random error, which is generally greater than the bias error for this situation. The bicubic spline interpolator gives the best results for both velocity and vorticity. The parabolic down (normalized) and Nogueira's weighting functions give the best results, but Nogueira's weighting function is much more susceptible to outliers in its calculations. Therefore, when using Nogueira's weighting function, it would be best to run a suitable outlier detection routine in between iterations to ensure that the algorithm converges on the correct result. For the current case, the best combination of parameters is found to be a window radius of 16 px, a seeding density of 5%, particles with a diameter of 4.9 px, a parabolic down (normalized) weighting window and bicubic spline interpolation. As noted earlier, if the fluid undergoes very large gradient changes and the velocity measurement is more important than vorticity, it may be beneficial to use interrogation windows with radii of 8 px.

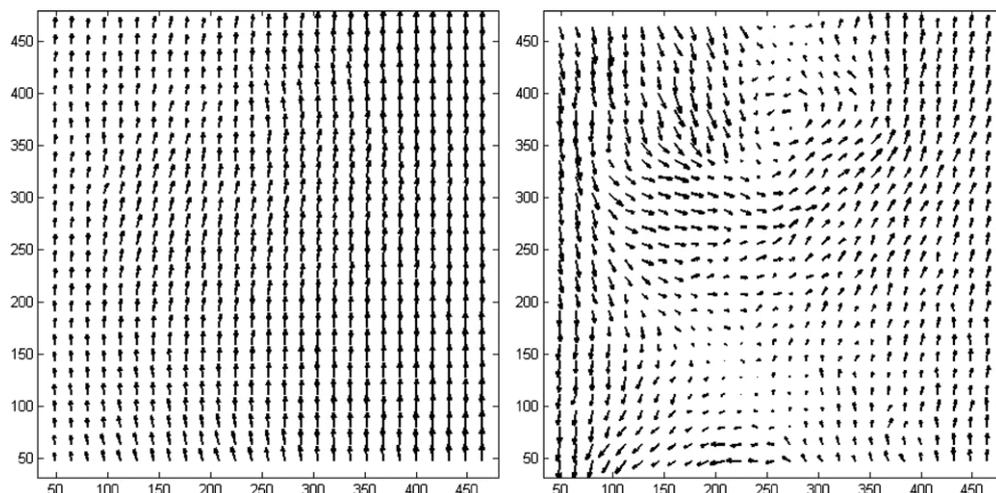
### 5.3. Experimental images

Using the same parameters that were found to be optimal in the Oseen vortex case, an image pair of a shear layer was processed. When windows with a radius of 16 px were used, the results did not converge properly, indicating that not enough data were contained in the window to allow the use of Nogueira's weighting function. Using a 32 px radius interrogation window allowed the use of Nogueira's weighting function while still giving good results. The results show that the current technique can be used not only in clean, synthetic images, but also on real experimental data.

### 5.4. Computational considerations

In this study, it is found for the uniform flow case that a bicubic interpolation produced better results than a bilinear interpolation, and that a bicubic spline interpolation reduced error further still. The uniform flow case showed a 40% total error reduction for the bicubic interpolation over the bilinear, and a further 30% reduction by using the bicubic spline interpolator. For the Oseen vortex case, only appreciable error reduction occurred within 50 px of the vortex center and was only apparent for the bicubic spline interpolator, decreasing the maximum velocity and vorticity total errors by 60% and 80%, respectively. Outside of 50 px, all three interpolators give the same result.

The increase in the accuracy of the bicubic and bicubic spline interpolators comes at the price of increased computational time. As an example, for the interrogation window radius of 8 px and the corresponding image size of  $184 \times 184$  px, the bilinear cases took approximately 1 min, the bicubic cases took approximately 30 min, and the bicubic spline cases took approximately 2.5–3 h real time, on the same computer. The attitude taken is similar to that of Fincham and Delerce (2000), in that the accuracy possible using more advanced interpolation algorithms is worth the computational expense.



**Figure 17.** (left) Results of the shear layer flow showing higher velocities on the right side of the image. (right) Velocity field with the mean flow subtracted, showing structures within the flow.

## 6. Conclusion

GIDPIV has been studied and the effects of interrogation window radii, particle image diameter, seeding density, weighting window choice and interpolation algorithm choice have been evaluated based on bias, random and total errors. This algorithm was applied to artificially generated images of both uniform flow and the flow around an Oseen vortex. For the uniform flow, larger interrogation windows tended to outperform the smaller windows, mainly due to the increased information they contained. It was found that lower seeding density (1% 1/px) resulted in lower bias errors for all but the smallest interrogation windows. Weighting windows that tended to keep more information (such as constant and parabolic down (shifted up)) resulted in a lower error than those that eliminated information (such as Nogueira and parabolic up), mainly because of the lack of data within an interrogation window. Of the interpolation algorithms tested, the more advanced schemes performed significantly better, with the bicubic spline resulting in the lowest error. A major difference found between GIDPIV and traditional PIV (Raffel et al 1998) was the optimum particle image diameter. For both the uniform and Oseen vortex flow, it was found that increasing particle image diameter resulted in a decreased error. Diameters up to 4.9 px were tested with no indication of a minimum of error, meaning that larger diameters than those tested may result in even lower errors. For the Oseen vortex case, interrogation windows with radii of 16 px proved optimal for measuring the vorticity of the flow and for most areas of the velocity. In areas of very high change in the velocity gradient, smaller windows tended to have less bias errors since the variation of the flow within the window was less. Overall however, the 16 px windows performed best. Here, the seeding density was found to have an opposite effect on error as it did in the uniform flow case (decreasing error with increasing density). Weighting window analysis showed that Nogueira's weighting function performed well, but included a high degree of scatter in the data. For this reason, and the fact that it performed slightly better near the vortex core, the parabolic down (normalized) was chosen as the optimal weighting window for GIDPIV. Investigation into the choice of interpolation algorithm showed effects only at radii below about 50 px. Inside this radius, the bicubic spline interpolator drastically decreases random error, and thus total error. The bicubic interpolator did not show a definitive improvement over the bilinear interpolator. GIDPIV has been shown to work on experimental images as well as synthetic images, showing structures within shear layer flow with properly chosen algorithm parameters. GIDPIV has been shown to perform well when compared with other advanced techniques with the benefit of calculating the shear and rotation in the flow directly via the optimization routine.

## Acknowledgment

The authors gratefully acknowledge the support of the National Institutes of Health (R01 RR023190-04 to JRH, DD and MG).

## References

- Astarita T 2007 Analysis of weighting windows for image deformation methods in PIV *Exp. Fluids* **42** 859–72
- Astarita T 2008 Analysis of velocity interpolation schemes for image deformation methods in PIV *Exp. Fluids* **45** 257–66
- Astarita T and Cardone G 2005 Analysis of interpolation schemes for image deformation methods in PIV *Exp. Fluids* **38** 233–43
- Dabiri D 2003 On the interaction of a vertical shear layer with a free surface *J. Fluid Mech.* **480** 217–32
- Dabiri D 2006 Cross-correlation digital particle image velocimetry—a review *Turbulencia* ed A S Freire, A Iiha and B Breidenthal (Rio de Janeiro: ABCM) pp 155–99
- Fincham A and Delerce G 2000 Advanced optimization of correlation imaging velocimetry algorithms *Exp. Fluids* **29** S13–22
- Fincham A and Spedding G 1997 Low cost, high resolution DPIV for measurement of turbulent flow *Exp. Fluids* **23** 449–62
- Gui L and Wereley T 2002 A correlation-based continuous window-shift technique to reduce the peak-locking effect in digital PIV image evaluation *Exp. Fluids* **32** 506–17
- Huang H T, Feidler H E and Wang J J 1993a Limitation and improvement of PIV: Part I. Limitation of conventional techniques due to deformation of particle image patterns *Exp. Fluids* **15** 168–74
- Huang H T, Feidler H E and Wang J J 1993b Limitation and improvement of PIV: Part II. Particle image distortion, a novel technique *Exp. Fluids* **15** 168–74
- Jambunathan K, Ju X Y, Dobbins B N and Ashforth-Frost S 1995 An improved cross correlation technique for particle image velocimetry *Meas. Sci. Technol.* **6** 507–14
- Keane R D and Adrian R J 1992 Theory of cross-correlation analysis of PIV images *Appl. Sci. Res.* **49** 191–215
- Kim B J and Sung H J 2006 A further assessment of interpolation schemes for window deformation in PIV *Exp. Fluids* **41** 499–511
- Lecuona A, Nogueira J, Rodriguez P A, Ruiz-Rivas U and Alfaro J 1999 Local field correction PIV: a superresolution technique *3rd Int. Workshop on PIV (Santa Barbara, CA, USA)*
- Mayer S 2002 A generalized processing technique in digital particle image velocimetry with direct estimation of velocity gradients *Exp. Fluids* **33** 443–57
- Meinhart C D, Wereley S T and Santiago J G 1999 PIV measurements of a microchannel flow *Exp. Fluids* **27** 414–9
- Meunier P and Leweke T 2003 Analysis and treatment of errors due to high velocity gradients in particle image velocimetry *Exp. Fluids* **35** 408–21
- Nogueira J, Lecuona A and Rodriguez P A 1999 Local field correction PIV: on the increase of accuracy of digital PIV systems *Exp. Fluids* **27** 107–16
- Nogueira J, Lecuona A and Rodriguez P A 2001 Local field correction PIV, implemented by means of simple algorithms, and multigrid versions *Meas. Sci. Technol.* **12** 1911–21
- Nogueira J, Lecuona A and Rodriguez P A 2005a Limits on the resolution of correlation PIV iterative methods. Fundamentals *Exp. Fluids* **39** 305–13
- Nogueira J, Lecuona A, Rodriguez P A, Alfaro J A and Acosta A 2005b Limits on the resolution of correlation PIV iterative methods. Practical implementation and design of weighting functions *Exp. Fluids* **39** 314–21
- Nogueira J, Lecuona A, Ruiz-Rivas U and Rodriguez P A 2002 Analysis and alternatives in two-dimensional multigrid particle image velocimetry methods: application of a dedicated weighting function and symmetric direct correlation *Meas. Sci. Technol.* **13** 963–74
- Powell M 1982 Extension to subroutine vf02ad *Lecture Notes in Control of Informatics Sciences* vol 38 (Berlin: Springer) pp 529–38

- Press W, Teukolsky S, Vetterling W and Flannery B 2002 *Recipes in C/C++: The Art of Scientific Computing* 2nd edn (Cambridge, UK: Cambridge University Press)
- Raffel M, Willert M and Kompenhans J 1998 *Particle Image Velocimetry: A Practical Guide* (Berlin: Springer)
- Ruan X *et al* 2001 Direct measurements of the vorticity field in digital particle images *Exp. Fluids*. **30** 696–704
- Scarano F 2002 Iterative image deformation methods in PIV *Meas. Sci. Technol.* **13** R1–19
- Scarano F and Riethmuller M L 2000 Advances in iterative multigrid PIV image processing *Exp. Fluids* **23** S51–60
- Wereley S T and Meinhart C D 2001 Second-order accurate particle image velocimetry *Exp. Fluids* **31** 258–68
- Westerweel J, Dabiri D and Gharib M 1997 The effect of a discrete window offset on the accuracy of cross-correlation analysis of digital PIV recordings *Exp. Fluids* **23** 20–8