

## Construction of 3D-PIV images from 2D measurements: an assessment

**Renata M. B. Chaves**

**Atila P. Silva Freire**

Mechanical Engineering Program (COPPE/UFRJ), C.P. 68503, 21945-970, Rio de Janeiro, Brazil

**Maria Helena Farias**

**Mikhail D. Mikhailov**

Scientific Division, Inmetro, 22.050-050, Rio de Janeiro, Brazil

**Abstract.** *A software package based on Mathematica is developed to find three-dimensional velocity fields from two-dimensional measurements in orthogonal planes located at different positions. The package can also be used to interpolate any three dimensional velocity field in a Cartesian coordinate system. This particular feature should permit Mathematica plotting functions to be used for representation of experimental PIV results. Some test results are shown for a spherical flow around a sink point. The velocity field of this flow is one-dimensional in a spherical coordinate system, but very complicated in a three-dimensional Cartesian coordinate system.*

**Keywords:** *three-dimensional PIV, Mathematica software, spherical flow.*

### 1. Introduction

Particle image velocimetry (PIV) has become a powerful tool for the investigation of complex flows. Despite the significant progress over the last two decades, the characterization of some three dimensional flows is still much troubled.

Because the classical PIV technique records the projected velocity vector field onto an illuminated cross-section of a seeded flow field, the out-of-plane velocity component is lost. To allow the acquisition of three-dimensional data, several schemes have been proposed (Prasad and Adrian (1993), Raffel et al. (1995), Elsinga et al. (2006), Dabiri (2008)). Perhaps the most praised way is to use a second camera to record the data from a different viewing angle. Then, the three-dimensional displacement field can be reconstructed from the two projected, planar fields. This technique is referred to as stereoscopic PIV. Holographic PIV uses the interference patterns of two laser beams of the same frequency but out of phase to visualize all the particles within a volume with their respective interference patterns. However, large difficulties in this technique occur when trying to extract the z-component (out of plane) of the particles. It is still in development but has so far not been reliably successful.

In any of the considered techniques, however, many complications arise. In addition to the complications resulting from the sometimes needed extra hardware, 3D-PIV calibration must be done with great care to ensure accurate results. The perspective distortion introduced by the angular offset camera configuration must be exactly corrected by a mapping function and this not always easily done.

The purpose of the present work is to investigate how accurately complex three dimensional velocity fields can be reconstructed from the interpolation of orthogonal two-dimensional planes. In a conventional two-dimensional PIV measuring system, the influence of the velocity component perpendicular to the light sheet on the determination of the in-plane particle displacement is not accounted for. The immediate consequence is that the in-plane particle displacement can be evaluated very simply by multiplying the image displacement by a constant factor. This procedure introduces a systematic error, but is justifiable for measurements of flow fields with weakly out-of-plane velocity components. With the present procedure such a problem does not occur.

The present work explores the possibilities of three-dimensional reconstruction of velocity fields from two-dimensional measured fields through a specially developed computational package. The package resorts to the software Mathematica<sup>TM</sup> due to its inherent powerful capabilities of deploying graphical output and interpolating functions. In previous works, Abe et al. (2000) and Yoon and Lee (2002) have compared three dimensional PIV measured data with two-dimensional results. Their works, however, do not propose a scheme of the same sort propose here.

In fact, before embarking onto a costly and time consuming experimental campaign, the present work explores the possibilities of the developed package by investigating spherical flows. The advantages of considering a bench-mark flow geometry are many. The two-dimensional data are easily obtained in any desired location and the reconstructed fields can be compared directly with the analytical solution. Spherical flows are very simply described in a spherical coordinate system, but yield a very complicated flow pattern in a three-dimensional Cartesian system.

A work of warning to readers is here necessary. The language used in the following developments is suited for readers who are familiar with the Mathematica syntax. Also, as mentioned before, the present focus of the paper is on three-dimensional interpolation of data obtained from a known analytical flow solution – spherical flows – so as to verify “exactly” the proposed interpolation procedure. Of course, this cannot be made with field PIV measurements.

The paper does present a consistent method of obtaining 3D fields from 2D measurements, a method particularly

useful for whole field techniques. In future contributions, an assessment of the technique will be made against 3D- and 2D-PIV field measurements.

## 2. The package

This section describes briefly the developed Mathematica package named “PIV2Dto3D”.

The package can be loaded by using the **Needs** function in any computer platform with installed Mathematica software system.

The present version of the package supports the functions **SphericalFlowVelocity**, **To3D**, **VelocityInterpolation**. These functions are documented as:

- **?SphericalFlowVelocity** SphericalFlowVelocity[Q, {x, x<sub>min</sub>, x<sub>max</sub>, Δx}, {y, y<sub>min</sub>, y<sub>max</sub>, Δy}, {z, z<sub>min</sub>, z<sub>max</sub>, Δz}]

gives the Cartesian coordinates {x, y, z} and the corresponding velocity components {u, v, w} of the radial velocity - Q/(4 π r<sup>2</sup>).

- **?To3D**

To3D[field1, field2] combines 2D field1 and field2 into 3D field. The field1 and the field2 have the form {{x, y, z, u, v, w}..}, where x, y, and z are the values of the coordinates at 3D

Cartesian mesh, u, v, and w are the value of corresponding velocity components one of which is zero. \nThe zero velocity component in fiel1 and field2 has to be different.

- **?VelocityInterpolation**

VelocityInterpolation[{{x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>, u<sub>1</sub>, v<sub>1</sub>, w<sub>1</sub>}, ...}] constructs a list of 3 interpolating functions of x, y, z for the velocity components u, v, and w by using the interpolation function options.

## 3. Spherical Flow

Spherical flow is a flow regime that occurs when the predominant flow pattern in a reservoir is toward a point.

In the petroleum industry, a spherical flow occurs for partial penetration and limited-entry completions. This flow regime is recognized as a -1/2 slope in the pressure derivative on the log-log diagnostic plot. Its presence enables determination of the spherical permeability. When spherical flow is followed by radial flow, both horizontal and vertical permeability can be quantified in a reservoir.

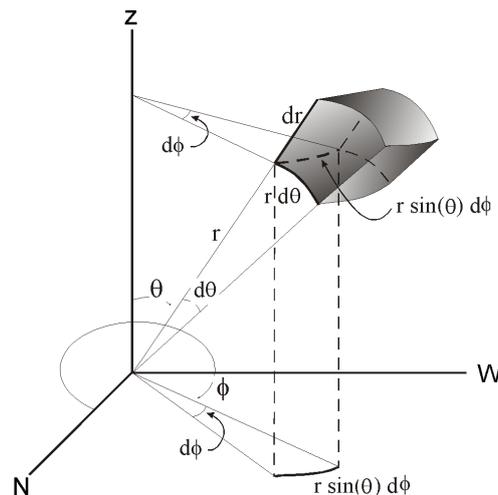


Figure 1. Cartesian {x(N), y(W), z(Z)} and spherical {r, θ, φ} coordinate systems. The ranges of the spherical coordinates are:  $0 \leq r < \infty$ ,  $0 \leq \theta \leq \pi$ , and  $-\pi \leq \phi \leq \pi$ .

This are the relations between the Cartesian and the spherical coordinates:

$$x = r \sin(\theta) \cos(\phi), \quad y = r \sin(\theta) \sin(\phi), \quad z = r \cos(\theta) \tag{1}$$

The spherical flow is described by Muskat (1982) by the model:

$$\Phi''(r) + \frac{2}{r} \Phi'(r) = 0, \Phi(r_0) = \Phi_0, \Phi(r_1) = \Phi_1 \quad (2)$$

where  $r_0$  is a small radius close to zero,  $r_1$  is a radius of the region under consideration,  $\Phi$  is the potential function.

The solution of the problem (2) is given by

$$\Phi(r) = \frac{(\Phi_1 - \Phi_0)}{\frac{1}{r_1} - \frac{1}{r_0}} \left( \frac{1}{r} - \frac{1}{r_0} \right) + \Phi_0 \quad (3)$$

The velocity is obtained by differentiation as

$$V(r) = -\Phi'(r) = \frac{(\Phi_1 - \Phi_0)}{\frac{1}{r_1} - \frac{1}{r_0}} \frac{1}{r^2} \quad (4)$$

The total flow through the system is given by

$$Q = 4\pi \frac{(\Phi_1 - \Phi_0)}{\frac{1}{r_0} - \frac{1}{r_1}} \quad (5)$$

Hence, velocity (4) can be rewritten as

$$V(r) = -\frac{Q}{4\pi r^2} \quad (6)$$

The components  $u(x)$ ,  $v(y)$ , and  $w(z)$  of the velocity (6) in Cartesian coordinate system are

$$u(x) = V(r) \sin(\theta) \cos(\phi), v(y) = V(r) \sin(\theta) \sin(\phi), w(z) = V(r) \cos(\theta) \quad (7)$$

The spherical flow velocity (7),  $\{u_i, v_i, w_i\}$ , at point with Cartesian coordinates  $\{x_i, y_i, z_i\}$  has to satisfy the conditions:

$$\frac{u_i}{x_i} = \frac{v_i}{y_i} = \frac{w_i}{z_i} \quad (8)$$

The function **SphericalFlowVelocity** gives the following radial velocities for  $Q=1$  at points defined by  $x_{\min}=0.2$  to  $x_{\max}=0.6$  step  $\Delta x=0.2$ ,  $y_{\min}=0.2$  to  $y_{\max}=0.6$  step  $\Delta y=0.2$ , and  $z_{\min}=0.2$  to  $z_{\max}=0.6$  step  $\Delta z=0.2$ .

Thus a working table (Table 1) can easily defined from

```
TableForm[xyzuvw = Flatten[SphericalFlowVelocity[1, {x, .2, .6, .2}, {y, .2, .6, .2}, {z, .2, .6, .2}, 2]]
```

This verify that the conditions (8),  $u_i/x_i = v_i/y_i = w_i/z_i$ , is fulfilled. It gives a list of sublists

$$u_i/x_i - v_i/y_i, u_i/x_i - w_i/z_i, \quad (9)$$

that becomes zero after applying the function **Chop**.

The documentation of **Chop** is given by `?Chop`

`Chop[expr]` replaces approximate real numbers in `expr` that are close to zero by the exact integer 0.

Use

```
Union[Map[Chop[{{#[[4]]/#[[1]] - #[[5]]/#[[2]], #[[4]]/#[[1]] - #[[6]]/#[[3]]}]&, xyzuvw]]
{{0, 0}}
```

Table 1. The coordinates  $x$ ,  $y$ ,  $z$  and the corresponding velocities  $u$ ,  $v$ ,  $w$  for spherical flow.

0.2	0.2	0.2	-0.382867	-0.382867	-0.382867
0.2	0.2	0.4	-0.135364	-0.135364	-0.270728
0.2	0.2	0.6	-0.0545307	-0.0545307	-0.163592
0.2	0.4	0.2	-0.135364	-0.270728	-0.135364
0.2	0.4	0.4	-0.0736828	-0.147366	-0.147366
0.2	0.4	0.6	-0.0379785	-0.075957	-0.113936
0.2	0.6	0.2	-0.0545307	-0.163592	-0.0545307
0.2	0.6	0.4	-0.0379785	-0.113936	-0.075957
0.2	0.6	0.6	-0.0240215	-0.0720644	-0.0720644
0.4	0.2	0.2	-0.270728	-0.135364	-0.135364
0.4	0.2	0.4	-0.147366	-0.0736828	-0.147366
0.4	0.2	0.6	-0.075957	-0.0379785	-0.113936
0.4	0.4	0.2	-0.147366	-0.147366	-0.0736828
0.4	0.4	0.4	-0.0957168	-0.0957168	-0.0957168
0.4	0.4	0.6	-0.0567658	-0.0567658	-0.0851487
0.4	0.6	0.2	-0.075957	-0.113936	-0.0379785
0.4	0.6	0.4	-0.0567658	-0.0851487	-0.0567658
0.4	0.6	0.6	-0.038559	-0.0578385	-0.0578385
0.6	0.2	0.2	-0.163592	-0.0545307	-0.0545307
0.6	0.2	0.4	-0.113936	-0.0379785	-0.075957
0.6	0.2	0.6	-0.0720644	-0.0240215	-0.0720644
0.6	0.4	0.2	-0.113936	-0.075957	-0.0379785
0.6	0.4	0.4	-0.0851487	-0.0567658	-0.0567658
0.6	0.4	0.6	-0.0578385	-0.038559	-0.0578385
0.6	0.6	0.2	-0.0720644	-0.0720644	-0.0240215
0.6	0.6	0.4	-0.0578385	-0.0578385	-0.038559
0.6	0.6	0.6	-0.0425408	-0.0425408	-0.0425408

#### 4. Velocity Interpolation

The simulated experimental results of PIV measurements – assigned to the symbol **xyzuvw** – are conveniently interpolated to yield the velocity vector in any desired points of the investigated region. Since the number of measured points is normally small, the default **InterpolationOrder** gives a warning message. To avoid this message the option **InterpolationOrder** could be specified as **{2,2,2}** or **2**.

This assigns to the symbols **U**, **V**, and **W** interpolating functions for the velocity components.

```
{U, V, W} = VelocityInterpolation[xyzuvw, InterpolationOrder → 2]
```

The output is

```
{InterpolatingFunction[{{0.2, 0.6}, {0.2, 0.6}, {0.2, 0.6}}, <>],  

InterpolatingFunction[{{0.2, 0.6}, {0.2, 0.6}, {0.2, 0.6}}, <>],  

InterpolatingFunction[{{0.2, 0.6}, {0.2, 0.6}, {0.2, 0.6}}, <>]}
```

The results are used to define **Velocity**[{**x**\_, **y**\_, **z**\_]:=U[**x**, **y**, **z**], V[**x**, **y**, **z**], W[**x**, **y**, **z**]

Next input line verify that the interpolated velocities coincide with the ones attached to the symbol **xyzuvw** and shown in Table 1.

```
Flatten[Table[Join[{x, y, z}, Velocity[{x, y, z}], {x, .2, .6, .2}, {y, .2, .6, .2}, {z, .2, .6, .2}], 2] == xyzuvw  

True
```

The obtained **Velocity** function computes the velocity components in any desired point of the investigated region. The vector and contour fields can be plotted through simple commands such as

```
VectorPlot3D[Velocity[{x, y, z}], {x, .2, .6}, {y, .2, .6}, {z, .2, .6}, AxesLabel → {"x", "y", "z"}]
```

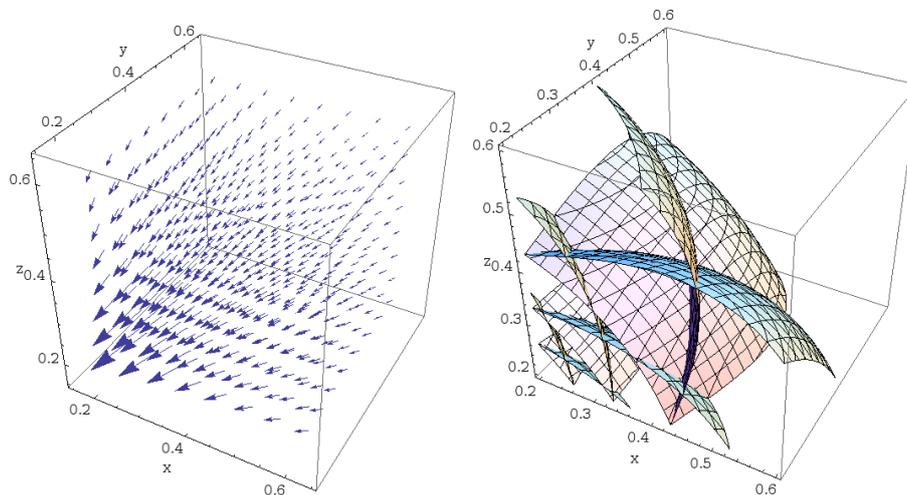


Figure 2. Spherical Velocity Field in the region by  $x_{min}=.2$  to  $x_{max}=.6$ ,  $y_{min}=.2$  to  $y_{max}=.6$ , and  $z_{min}=.2$  to  $z_{max}=.6$ . 2a: vector field. 2b: contour surfaces.

#### 4.1 3D from 2D velocity fields

Suppose that we are able to measure in **x-y** plane at **z=.2**, **.4**, and **.6** the velocities **u** and **v**. The unknown velocity **w** in **z** direction is replaced by zero.

The 2D simulated measurements at **z=0.2** (Table 2) can be obtained from

```
TableForm[z02 = Map[ReplacePart[#, 6 → 0]&, Flatten[SphericalFlowVelocity[1, {x, .2, .6, .2}, {y, .2, .6, .2}, {z, .2, .2}], 2]]]
```

Table 2. 2D simulated measurements at **z=.2**

0.2	0.2	0.2	-0.382867	-0.382867	0
0.2	0.4	0.2	-0.135364	-0.270728	0
0.2	0.6	0.2	-0.0545307	-0.163592	0
0.4	0.2	0.2	-0.270728	-0.135364	0
0.4	0.4	0.2	-0.147366	-0.147366	0
0.4	0.6	0.2	-0.075957	-0.113936	0
0.6	0.2	0.2	-0.163592	-0.0545307	0
0.6	0.4	0.2	-0.113936	-0.075957	0
0.6	0.6	0.2	-0.0720644	-0.0720644	0

Table 3. 2D simulated measurements at **z=.4**

0.2	0.2	0.4	-0.135364	-0.135364	0
0.2	0.4	0.4	-0.0736828	-0.147366	0
0.2	0.6	0.4	-0.0379785	-0.113936	0
0.4	0.2	0.4	-0.147366	-0.0736828	0
0.4	0.4	0.4	-0.0957168	-0.0957168	0
0.4	0.6	0.4	-0.0567658	-0.0851487	0
0.6	0.2	0.4	-0.113936	-0.0379785	0
0.6	0.4	0.4	-0.0851487	-0.0567658	0
0.6	0.6	0.4	-0.0578385	-0.0578385	0

The 2D simulated measurements at **z=0.4** (Table 3) can be obtained from  
**TableForm[z04 = Map[ReplacePart[#, 6 → 0]&, Flatten[SphericalFlowVelocity[1, {x, .2, .6, .2}, {y, .2, .6, .2}, {z, .4, .4}], 2]]]**

The 2D simulated measurements at **z=0.6** (Table 4) can be obtained from  
**TableForm[z06 = Map[ReplacePart[#, 6 → 0]&, Flatten[SphericalFlowVelocity[1, {x, .2, .6, .2}, {y, .2, .6, .2}, {z, .6, .6}], 2]]]**

Table 4. 2D simulated measurements at **z=.6**

0.2	0.2	0.6	-0.0545307	-0.0545307	0
0.2	0.4	0.6	-0.0379785	-0.075957	0
0.2	0.6	0.6	-0.0240215	-0.0720644	0
0.4	0.2	0.6	-0.075957	-0.0379785	0
0.4	0.4	0.6	-0.0567658	-0.0567658	0
0.4	0.6	0.6	-0.038559	-0.0578385	0
0.6	0.2	0.6	-0.0720644	-0.0240215	0
0.6	0.4	0.6	-0.0578385	-0.038559	0
0.6	0.6	0.6	-0.0425408	-0.0425408	0

The above "measurements" assigned to the symbols **z02**, **z04**, and **z06** are joined and sorted in the list **xyzuv0** which gives the coordinates **x**, **y**, **z**, and the velocity components **u**, **v**, **0**. Since the last component **w** is not measured it is replaced by zero.

The complete 2D simulated measurements (Table 5, Fig. (4)) at all three planes are obtained with  
**TableForm[xyzuv0 = Sort[Join[z02, z04, z06]]]**

Note that the above list **xyzuv0** is equivalent to the **3D** list **xyzuvw** in which the velocity components **w** are replaced by zeros.

**xyzuv0==Map[ReplacePart[#, 6 → 0]&, xyzuvw]**  
 True

Table 5. 2D simulated measurements at  $z=0.2, 0.4$  and  $0.6$

0.2	0.2	0.2	-0.382867	-0.382867	0
0.2	0.2	0.4	-0.135364	-0.135364	0
0.2	0.2	0.6	-0.0545307	-0.0545307	0
0.2	0.4	0.2	-0.135364	-0.270728	0
0.2	0.4	0.4	-0.0736828	-0.147366	0
0.2	0.4	0.6	-0.0379785	-0.075957	0
0.2	0.6	0.2	-0.0545307	-0.163592	0
0.2	0.6	0.4	-0.0379785	-0.113936	0
0.2	0.6	0.6	-0.0240215	-0.0720644	0
0.4	0.2	0.2	-0.270728	-0.135364	0
0.4	0.2	0.4	-0.147366	-0.0736828	0
0.4	0.2	0.6	-0.075957	-0.0379785	0
0.4	0.4	0.2	-0.147366	-0.147366	0
0.4	0.4	0.4	-0.0957168	-0.0957168	0
0.4	0.4	0.6	-0.0567658	-0.0567658	0
0.4	0.6	0.2	-0.075957	-0.113936	0
0.4	0.6	0.4	-0.0567658	-0.0851487	0
0.4	0.6	0.6	-0.038559	-0.0578385	0
0.6	0.2	0.2	-0.163592	-0.0545307	0
0.6	0.2	0.4	-0.113936	-0.0379785	0
0.6	0.2	0.6	-0.0720644	-0.0240215	0
0.6	0.4	0.2	-0.113936	-0.075957	0
0.6	0.4	0.4	-0.0851487	-0.0567658	0
0.6	0.4	0.6	-0.0578385	-0.038559	0
0.6	0.6	0.2	-0.0720644	-0.0720644	0
0.6	0.6	0.4	-0.0578385	-0.0578385	0
0.6	0.6	0.6	-0.0425408	-0.0425408	0

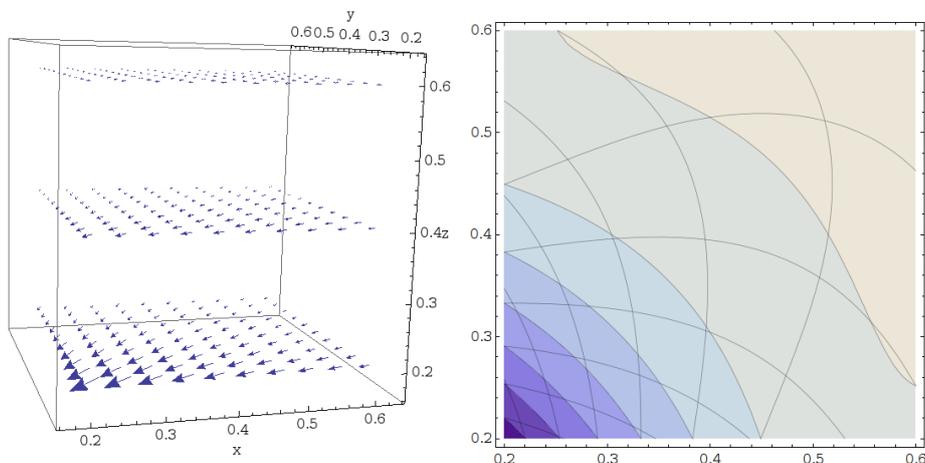


Figure 3. Reduced velocity fields in the various  $z$ -planes. 3a: vector field. 3b: contour lines.

Suppose now that we are able to measure in **x-z** plane at **y=0.2, 0.4, and 0.6** and obtain the following 2D simulated measurements. The unknown velocity in **y** direction is replaced by zero.

The 2D simulated measurements at **y=0.2** (Table 6) can be obtained from

**TableForm[y02 = Map[ReplacePart[#, 5 → 0]&, Flatten[SphericalFlowVelocity[1, {x, .2, .6, .2}, {y, .2, .2}, {z, .2, .6, .2}], 2]]]**

Table 6. 2D simulated measurements at **y=0.2**

0.2	0.2	0.2	-0.382867	0	-0.382867
0.2	0.2	0.4	-0.135364	0	-0.270728
0.2	0.2	0.6	-0.0545307	0	-0.163592
0.4	0.2	0.2	-0.270728	0	-0.135364
0.4	0.2	0.4	-0.147366	0	-0.147366
0.4	0.2	0.6	-0.075957	0	-0.113936
0.6	0.2	0.2	-0.163592	0	-0.0545307
0.6	0.2	0.4	-0.113936	0	-0.075957
0.6	0.2	0.6	-0.0720644	0	-0.0720644

The 2D simulated measurements at **y=0.4** (Table 7) can be obtained from

**TableForm[y04 = Map[ReplacePart[#, 5 → 0]&, Flatten[SphericalFlowVelocity[1, {x, .2, .6, .2}, {y, .4, .4}, {z, .2, .6, .2}], 2]]]**

Table 7. 2D simulated measurements at **y=0.4**

0.2	0.4	0.2	-0.135364	0	-0.135364
0.2	0.4	0.4	-0.0736828	0	-0.147366
0.2	0.4	0.6	-0.0379785	0	-0.113936
0.4	0.4	0.2	-0.147366	0	-0.0736828
0.4	0.4	0.4	-0.0957168	0	-0.0957168
0.4	0.4	0.6	-0.0567658	0	-0.0851487
0.6	0.4	0.2	-0.113936	0	-0.0379785
0.6	0.4	0.4	-0.0851487	0	-0.0567658
0.6	0.4	0.6	-0.0578385	0	-0.0578385

This are the 2D simulated measurements at **y=0.6** (Table 8):

**TableForm[y06 = Map[ReplacePart[#, 5 → 0]&, Flatten[SphericalFlowVelocity[1, {x, .2, .6, .2}, {y, .6, .6}, {z, .2, .6, .2}], 2]]]**

Table 8. 2D simulated measurements at **y=0.6**

0.2	0.6	0.2	-0.0545307	0	-0.0545307
0.2	0.6	0.4	-0.0379785	0	-0.075957
0.2	0.6	0.6	-0.0240215	0	-0.0720644
0.4	0.6	0.2	-0.075957	0	-0.0379785
0.4	0.6	0.4	-0.0567658	0	-0.0567658
0.4	0.6	0.6	-0.038559	0	-0.0578385
0.6	0.6	0.2	-0.0720644	0	-0.0240215
0.6	0.6	0.4	-0.0578385	0	-0.038559
0.6	0.6	0.6	-0.0425408	0	-0.0425408

The above “measurements” assigned to the symbols **y02, y04, and y06** are joined and sorted in the list **xyzu0w** which gives the coordinates **x, y, z**, and the velocity components **u, 0, w**. Since the second component **v** is not measured it is replaced by zero.

The simulated velocity measurements at y-planes are shown in (Table 9) and Fig.(4).

**TableForm[xyzu0w = Sort[Join[y02, y04, y06]]]**

Table 9. 2D simulated measurements at **y=0.2, 0.4 and 0.6**

0.2	0.2	0.2	-0.382867	0	-0.382867
0.2	0.2	0.4	-0.135364	0	-0.270728
0.2	0.2	0.6	-0.0545307	0	-0.163592
0.2	0.4	0.2	-0.135364	0	-0.135364
0.2	0.4	0.4	-0.0736828	0	-0.147366
0.2	0.4	0.6	-0.0379785	0	-0.113936
0.2	0.6	0.2	-0.0545307	0	-0.0545307
0.2	0.6	0.4	-0.0379785	0	-0.075957
0.2	0.6	0.6	-0.0240215	0	-0.0720644
0.4	0.2	0.2	-0.270728	0	-0.135364
0.4	0.2	0.4	-0.147366	0	-0.147366
0.4	0.2	0.6	-0.075957	0	-0.113936
0.4	0.4	0.2	-0.147366	0	-0.0736828
0.4	0.4	0.4	-0.0957168	0	-0.0957168
0.4	0.4	0.6	-0.0567658	0	-0.0851487
0.4	0.6	0.2	-0.075957	0	-0.0379785
0.4	0.6	0.4	-0.0567658	0	-0.0567658
0.4	0.6	0.6	-0.038559	0	-0.0578385
0.6	0.2	0.2	-0.163592	0	-0.0545307
0.6	0.2	0.4	-0.113936	0	-0.075957
0.6	0.2	0.6	-0.0720644	0	-0.0720644
0.6	0.4	0.2	-0.113936	0	-0.0379785
0.6	0.4	0.4	-0.0851487	0	-0.0567658
0.6	0.4	0.6	-0.0578385	0	-0.0578385
0.6	0.6	0.2	-0.0720644	0	-0.0240215
0.6	0.6	0.4	-0.0578385	0	-0.038559
0.6	0.6	0.6	-0.0425408	0	-0.0425408

Note that the obtained data list **xyzu0w** is equivalent to the 3D list **xyzuvw** in which the velocity components **v** are replaced by zeros.

In fact,

**xyzu0w==Map[ReplacePart[#, 5 → 0]&, xyzuvw]**  
 True

Finally, the function **To3D** use the 2D lists **xyzuv0** and **xyzu0w** to give the 3D spherical field assigned to the symbol **xyzuvw**, that is,

**To3D[xyzuv0, xyzu0w] == xyzuvw**  
 True

Suppose that we are able to measure in **y-z** plane at **x=0.2, 0.4, and 0.6** the velocities **v** and **w**. The unknown velocity **u** in **x** direction is replaced by zero. The command below is used to implement this

**xyz0vw = Map[ReplacePart[#, 4 → 0]&, xyzuvw];**

Suppose that one of the measurements **xyzuv0** or **xyzu0w** is replaced by **xyz0vw**. Than the function **To3D** again gives correctly 3D spherical fields:

**To3D[xyz0vw, xyzu0w] == xyzuvw**

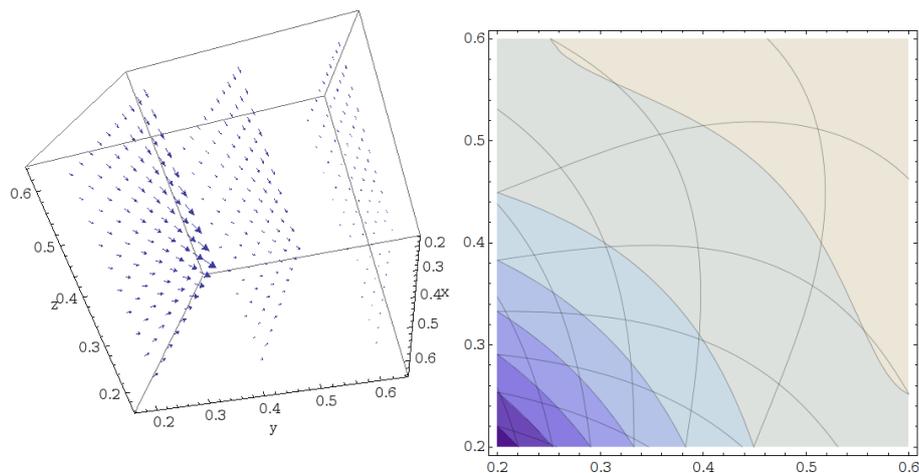


Figure 4. Reduced velocity fields in the various y-planes. 3a: vector field. 3b: contour lines.

True

`To3D[xyz0vw, xyzuv0] == xyzuvw`

True

## 5. Conclusion

The successful finding of velocity components for three-dimensional spherical flow encourages the authors to apply the presently developed *Mathematica* package to determine three-dimensional velocity fields from experimentally obtained two-dimensional PIV measurements in orthogonal planes located at different positions. Current work in this direction is currently under way and will be reported soon.

## 6. Acknowledgements

The financial support provided by FAPERJ is greatly acknowledged by MDM. APSF is grateful to the Brazilian National Research Council (CNPq) for the award of a Research Fellowship (Grant No 306977/2006-0). The work was financially supported by Petrobras through Grant Coppetec No 11298, by the CNPq through Grant No 476091/2007 and by the Rio de Janeiro Research Foundation (FAPERJ) through Grant E-26/170.005/2008.

## 7. References

- M. Abe, E.K. Longmire, K. Hishida, M. Maeda, 2000, A Comparison of 2D and 3D PIV Measurements in an Oblique Jet, *Journal of Visualization*.
- D. Dabiri, 2008, Digital particle image thermometry/velocimetry: a review. *Exp. Fluids*, DOI 10.1007/s00348-008-0590-5.
- G.E. Elsinga, F. Scarano, B. Wieneke, B.W. van Oudheusden, 2006, Tomographic particle image velocimetry, *Exp. Fluids*, 41:933:974.
- M. Muskat, 1982, *The Flow Of Homogeneous Fluids Through Porous Media*, International Human Resources Development Corporation, Boston, ISBN: 0-934634-16-5
- A.K. Prasad and Adrian, R.J., 1993, Stereoscopic particle image velocimetry applied to liquid flows, *Exp.Fluids*, 15:49-60
- M. Raffel, M. Gharib, O. Ronnerberg, J. Kompenhans, 1995, Feasibility study of three-dimensional PIV correlating images of particles within parallel light sheet planes, *Exp. Fluids*, 19:69-77
- J.H. Yoon, S.J. Lee, 2002, Direct comparison of 2D PIV and stereoscopic PIV measurements *Measurement Science and Technology*, 13:1631–1642.