High performance computing for a 3-D optical diffraction tomographic application in fluid velocimetry

Julia Lobera,[1,*] Gloria Ortega,[2] Inmaculada García,[3] María del Pilar Arroyo,[4] and Ester M. Garzón,[2]

[1]*Centro Universitario de la Defensa de Zaragoza, Ctra. Huesca s/n, 50090 Zaragoza, Spain*
[2]*Informatics Dpt., Agrifood Campus of Int. Excellence (ceiA3), Univ. of Almería, Ctra. Sacramento s/n, 04120 Almería, Spain*
[3]*Dpt. Computer Architecture, E.T.S.I. Informática, Univ. of Málaga, 29071, Málaga, Spain*
[4]*Aragón Institute of Engineering Research (I3A), Univ. of Zaragoza, 50009, Zaragoza, Spain*
[*]*jlobera@unizar.es*

Abstract: Optical Diffraction Tomography has been recently introduced in fluid velocimetry to provide three dimensional information of seeding particle locations. In general, image reconstruction methods at visible wavelengths have to account for diffraction. Linear approximation has been used for three-dimensional image reconstruction, but a non-linear and iterative reconstruction method is required when multiple scattering is not negligible. Non-linear methods require the solution of the Helmholtz equation, computationally highly demanding due to the size of the problem. The present work shows the results of a non-linear method customized for spherical particle location using GPU computing and a made-to-measure storing format.

OCIS codes: (090.1995) Digital holography; (110.1758) Computational imaging; (290.4210) Multiple scattering; (100.3010) Image reconstruction techniques.

References and links

1. M. P. Arroyo and K. D. Hinsch, "Recent developments of PIV towards 3D measurements," in *Particle Image Velocimetry, volume 112 of Topics in Applied Physics*, A. Schroder and C.E. Willert, eds. (Springer, 2008), pp. 127–154.
2. J. M. Coupland and N. A. Halliwell, "Particle image velocimetry: three-dimensional fluid velocity measurements using holographic recording and optical correlation," Appl. Opt. **31**(8), 1005–1007 (1992).
3. J. Katz and J. Sheng, "Applications of Holography in Fluid Mechanics and Particle Dynamics," Annu. Rev. Fluid Mech. **42**(1), 531–555 (2010).
4. W. D. Koek, N. Bhattacharya, J. J. M. Braat, T. A. Ooms, and J. Westerweel, "Influence of virtual images on the signal-to-noise ratio in digital in-line particle holography," Opt. Express **13**(7), 2578–2589 (2005).
5. J. Soria and C. Atkinson, "Towards 3C-3D digital holographic fluid velocity vector field measurement-tomographic digital holographic PIV (Tomo-HPIV)," Meas. Sci. Technol. **19**(7), 074002 (2008).
6. J. Sheng, E. Malkiel, and J. Katz, "Single beam two-views holographic particle image velocimetry," Appl. Opt. **42**(2), 235–250 (2003).
7. J. Lobera and J. M. Coupland, "Optical diffraction tomography in fluid velocimetry: the use of a priori information," Meas. Sci. Technol. **19**(7), 074013 (2008).
8. E. Wolf, "Three-dimensional structure determination of semi-transparent objects from holographic data," Opt. Commun. **1**(4), 153–156 (1969).
9. K. Belkebir and A. Sentenac, "High-resolution optical diffraction microscopy," J. Opt. Soc. Am. A **20**(7), 1223–1229 (2003).
10. P. C. Chaumet, A. Sentenac, K. Belkebir, G. Maire, and H. Giovannini, "Improving the resolution of grating-assisted optical diffraction tomography using a priori information in the reconstruction procedure," J. Mod. Opt. **57**(9), 798–808 (2010).
11. F. Soulez, L. Denis, C. Fournier, É. Thiébaut, and C. Goepfert, "Inverse-problem approach for particle digital holography: accurate location based on local optimization," J. Opt. Soc. Am. A **24**(4), 1164–1171 (2007).
12. China's TianhE-2 Supercomputer Maintains Top Spot on 42nd. TOP500 List. June 2014.

13. G. Ortega, J. Lobera, M. P. Arroyo, I. García, and E. M. Garzón, "High performance computing for optical diffraction tomography". *In Proceedings of High Performance Computing Simulation*, (IEEE, 2012), pp. 195–201.
14. G. Ortega, J. Lobera, I. García, M. P. Arroyo, and E. M. Garzón, "Parallel resolution of the 3D Helmholtz equation based on multi-GPU clusters," Concurr. Comput. (2014), doi:10.1002/cpe.3212.
15. Mathworks. MATLAB with MEX Files. *http://www.mathworks.es/es/help/distcomp/run-mex-functions-containing-cuda-code.html.*
16. M. N. O. Sadiku, *Numerical Techniques in Electromagnetics* (CRC Press, 2001).
17. J. M. Coupland and J. Lobera, "Holography, tomography and 3D microscopy as linear filtering," Meas. Sci. Technol. **19**(7), 074012 (2008).
18. F. Ihlenburg and I. Babuska, "Finite element solution of the Helmholtz equation with high wave number Part I: The H-version of the FEM," Comput. Math. Appl. **30**(9), 9–37 (1995).
19. I. M. Babuska and S. A. Sauter, "Is the pollution effect of the FEM Avoidable for the Helmholtz Equation considering high wave numbers?" SIAM Rev. **42**(3), 451–484 (2000).
20. Y. Saad, *Iterative Methods for Sparse Linear Systems* (SIAM, 2003).
21. G. Ortega, E. M. Garzón, F. Vázquez, and I. García, "The BiConjugate gradient method on GPUs," J. Supercomput. **64**(1), 49–58 (2013).
22. G. Ortega, E. M. Garzón, F. Vázquez, and I. García, "Exploiting the regularity of differential operators to accelerate solutions of PDEs on GPUs," *In Proceedings of the 11th International Computational and Mathematical Methods in Science and Engineering*, J. Vigo-Aguiar, ed. (CMMSE, 2011), pp. 908–917.
23. *NVIDIA. CUDA Toolkit.* https://developer.nvidia.com/cuda-toolkit
24. CUBLAS user guide (du-06702–001 v5.5). *http://docs.nvidia.com/cuda/pdf/CUBLAS_Library.pdf.*
25. G. Ortega, I. García, and E. M. Garzón, "A Hybrid Approach for Solving the 3D Helmholtz Equation on Heterogeneous Platforms," *In Euro-Par 2013: Parallel Processing Workshops, volume 8374 of LNCS*, D. an Mey et al., eds. (Springer, 2014), pp. 198–207.

1. Introduction

Holographic Particle Image Velocimetry (HPIV) provides simultaneous three components, three-dimensional (3C-3D) velocity measurements of a seeded fluid flow [1–3]. The classical analysis of HPIV recordings assumes that the particle illuminating and diffracted beams do not suffer multiple scattering. In practice, however, multiple scattering effects are common and increase background noise, which decrease the number of velocity vectors that can be retrieved from a given flow field [4]. Tomographic methods using several recordings from different observation directions have been proposed to mitigate this problem [5, 6].The application of Optical Diffraction Tomography (ODT) in HPIV, and more specifically the Non-Linear ODT, would improve the spatial resolution [7].

If the linear approximation is assumed, the spectral components of the field scattered by the object are directly related to the spectral components of the object refractive index field [8]. For the Non-Linear ODT approach [7, 9], the image reconstruction is considered to be the solution that better explains the scattered far field but it requires a bigger computational cost. In the implementation of this optimization method, the Helmholtz equation needs to be solved for a known refractive index distribution and an illuminating field - the forward problem. Appropriate sampling is roughly a tenth of the wavelength, which makes the computational requirements very demanding even for small volume flows. Thus, the performance of the Non-Linear ODT (NLODT) in HPIV is determined by the computing strategy, and this is particularly true for its implementation to 3D problems. Both the use of a priori information concerning the object and High Performance Computing (HPC) could improve this performance. The use of a priori information can reduce instability in optimization and computational time [7, 10]. In fluid velocimetry this information is on the diameter and optical properties of the seeding particles, effectively reducing the particle imaging reconstruction problem to a particle location problem, as showed in [7, 11].

Nowadays, HPC platforms are characterized by the heterogeneity of their resources. Most of the modern supercomputers consist of clusters of multi-core nodes which include accelerator devices such as GPUs [12]. Earlier works have shown the parallel computation capability of GPUs in performing ODT models [13, 14]. In this paper, we will focus our attention on the combination of MATLAB with GPU devices in order to offload part of the computation and improving the global performance. In particular, we will discuss an

implementation of our NLODT model at a source-level MATLAB compiler calling MEX-files for using GPU routines [15], which is also experimentally evaluated.

Even though, for a biological flow of (10mm)³ a multiscale strategy will be needed. In a first step, a gross image can be obtained with Linear ODT (LODT), where the brightest spots will determine the most likely particle locations. Then, a finer resolution could be obtained with a local NLODT that can unravel the particle position within cluster of particles.

In this paper some numerical experiments have been carried out to locate 2µm particles within clusters of roughly (10µm)³. A comparison between LODT and NLODT results will show the relevance of the multiple scattering between neighbour particles and the multiple scattering due to the particle finite size. In particular, we make the following contributions: (1) development and validation of a NLODT 3D model for the location of particles (NLODT-P) when different source of multiple scattering are present; and (2) an efficient exploitation of the parallel computing power of the GPU devices for performing optical diffraction tomography image reconstruction.

The paper is organized as follows. Section 2 studies and analyzes the proposed model (NLODT-P). Section 3 is devoted to discussing the implementation based on GPU. Section 4 validates the model with two numerical experiments. Finally, Section 5 summarizes the main conclusions and future works.

2. Description of NLODT-P model

In fluid velocimetry the flow is seeded by small particles with a different refractive index. A typically coherent source is used to illuminate the flow, and this illuminating beam is scattered by the seeding particles marking their position at one time instant. To determine the velocity field of the flow, two consecutive recordings are required. In particular, in this work we will focus on the combination of several simultaneous coherent recordings to recover the position of each particle on a certain volume of interest at one time. The same procedure can be done with any set of simultaneous recordings and standard PIV analysis can be used afterward to determine the flow velocity field.

NLODT aims to find the refractive index distribution $n(\mathbf{r})$ that could explain simultaneously any of the measured scattering fields $E_m(\mathbf{r})$. It requires the expected measured field, given a known illuminating field and an estimation of $n(\mathbf{r})$, to be computed. This is a forward problem, which consists of computing the scattered field $E_{sth}(\mathbf{r})$ by a known object and an illuminating or reference beam $E_r(\mathbf{r})$. According to scalar diffraction theory the (complex) amplitude of a monochromatic electric field, $E(\mathbf{r}) = E_{sth}(\mathbf{r}) + E_r(\mathbf{r})$, propagating in a medium of (complex) refractive index, $n(\mathbf{r})$, obeys the Helmholtz equation [16] which can be rewritten as follows:

$$\left(\nabla^2 + k_0^2 n^2 (\mathbf{r}) \right) E_{sth} (\mathbf{r}) = f (\mathbf{r}) E_r (\mathbf{r}), \tag{1}$$

where $f(\mathbf{r})$ is the scattering potential, defined by $f(\mathbf{r}) = -k_0^2 \left(n^2 (\mathbf{r}) - 1 \right)$. Although there is no a linear relation between the refractive index and the scattering potential, both of them can be taken as a good representation of the object. Let us remark that, $E_m(\mathbf{r})$ is a far-field measurement and therefore, the forward solution $E_{sth}(\mathbf{r})$ needs to be filtered to remove the non-propagating waves [17], thus obtaining the expected measured field, $E_c(\mathbf{r})$. The optimization problem involved in the tomographic reconstruction consists in a minimization of the cost function defined by the square root difference between the measured and the computed field. For any hologram there is a separate contribution to the cost function denoted by the index i thereinafter:

$$\cos t(f) = \sum_i \left| E_m^i (\mathbf{r}) - E_c^i (f, \mathbf{r}) \right|^2 . \tag{2}$$

This is a non-linear optimization problem that would be addressed by a modified Conjugated Gradient Method (CGM) [7]. The search direction at any iteration is the negative gradient of the cost function, denoted by g(**r**). This gradient can be taken as the increment $\Delta f(\mathbf{r})$ that should be added to the available estimation of $f(\mathbf{r})$ to reduce the cost function and therefore to better justify the measurements. For the initial iteration, the estimation of the scattering potential $f(\mathbf{r})$ is null in the whole volume considered (i.e. free space). Subsequently the gradient can be taken as the Linear ODT image of the object. This image is typically a smooth distribution with several local maxima, even when some sharp refractive index changes are expected. In particular, in fluid velocimetry applications, the object is a sparse distribution of particles of known refractive index and shape. Subsequently a relatively small matrix *PL* that stores the particle location can describe the 3D refractive index field. The gradient will be used to obtain the *PL* matrix instead of being used as a reliable image of the particle field.

According to the previous considerations we propose the model referred as NLODT-P which is composed by (1) an initial iteration, where the location of the first particle is determined, and (2) an iterative process, where the remaining particles are located. The initial iteration does contain similar steps to those in the subsequent iterations, but no special computing resources are required, therefore it is considered separately. The NLODT-P algorithm used is outlined in Fig. 1 and the main details for every step are described below.
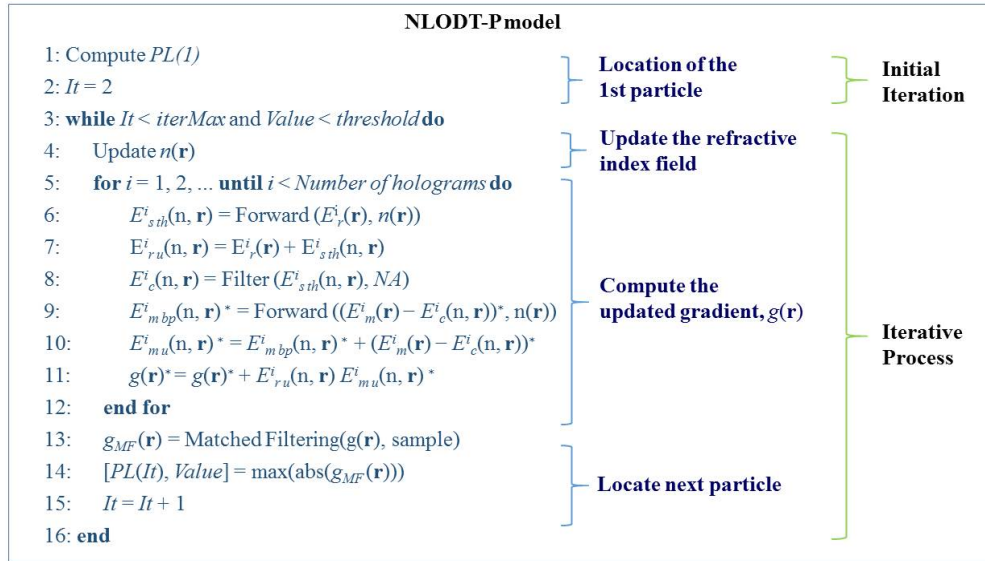

NLODT-P model

1: Compute *PL(1)*

2: *It* = 2 ⎤ Location of the ⎤ **Initial**
 ⎦ 1st particle ⎦ **Iteration**

3: **while** *It < iterMax* and *Value < threshold* **do**

4:   Update *n*(**r**) ⎤ **Update the refractive**
 ⎦ **index field**

5:   **for** *i* = 1, 2, ... **until** *i < Number of holograms* **do**

6:     $E^i_{s\,th}(n, \mathbf{r}) = \text{Forward}(E^i_r(\mathbf{r}), n(\mathbf{r}))$

7:     $E^i_{r\,u}(n, \mathbf{r}) = E^i_r(\mathbf{r}) + E^i_{s\,th}(n, \mathbf{r})$

8:     $E^i_c(n, \mathbf{r}) = \text{Filter}(E^i_{s\,th}(n, \mathbf{r}), NA)$ **Compute the**

9:     $E^i_{m\,bp}(n, \mathbf{r})^* = \text{Forward}((E^i_m(\mathbf{r}) - E^i_c(n, \mathbf{r}))^*, n(\mathbf{r}))$ **updated gradient,** g(r)   ⎤ **Iterative**

10:    $E^i_{m\,u}(n, \mathbf{r})^* = E^i_{m\,bp}(n, \mathbf{r})^* + (E^i_m(\mathbf{r}) - E^i_c(n, \mathbf{r}))^*$ ⎦ **Process**

11:    $g(\mathbf{r})^* = g(\mathbf{r})^* + E^i_{r\,u}(n, \mathbf{r})\, E^i_{m\,u}(n, \mathbf{r})^*$

12:   **end for**

13:   $g_{MF}(\mathbf{r}) = \text{Matched Filtering}(g(\mathbf{r}), \text{sample})$

14:   $[PL(It), Value] = \max(\text{abs}(g_{MF}(\mathbf{r})))$ **Locate next particle**

15:   *It* = *It* + 1

16: **end**

Fig. 1. Algorithm of the NLODT-P model.

Step 1. Location of the 1st particle

The gradient of the cost function can be expressed as the sum of the simulated Bragg holograms between the illuminating field and the back propagated measured field [7]. In particular, for the initial iteration, the illuminating field $E_r^i(\mathbf{r})$ and the back propagation of the measured field $E_m^i(\mathbf{r})$ are considered undisturbed. Thus the gradient is essentially the First Born Approximation of the scattering potential, i.e. the LODT image:

$$g^*(\mathbf{r}) = \sum_i E_m^{i\,*}(\mathbf{r})\, E_r^i(\mathbf{r}). \qquad (3)$$

Conjugated value is represented by * on equations. The particle location could be obtained from the position of the maximum of the absolute value of $g(\mathbf{r})$. A better performance of the model can be obtained if the gradient is previously filtered in order to find the pattern

produced by a sample particle. The LODT image obtained from an isolated particle is considered as matched-filter (*sample*) and computed separately.

*Step 2. Update the refractive index field, n(**r**)*

From the a priori knowledge of the object in fluid velocimetry, we assume that the refractive index field can only take: (1) the refractive index of the seeding particles within a spherical region around any located particle; and (2) the fluid refractive index in the remaining voxels of the volume of interest. So, when a new particle is located, $n(\mathbf{r})$ is updated accordingly.

*Step 3. Compute the updated gradient, g(**r**)*

Once an estimation of the scattering potential is available, a similar expression to Eq. (3) can be obtained for the new gradient $g(\mathbf{r})$. However, the meaning of both interfering fields and its computational strategy have changed. Figure 2 illustrates schematically the process.
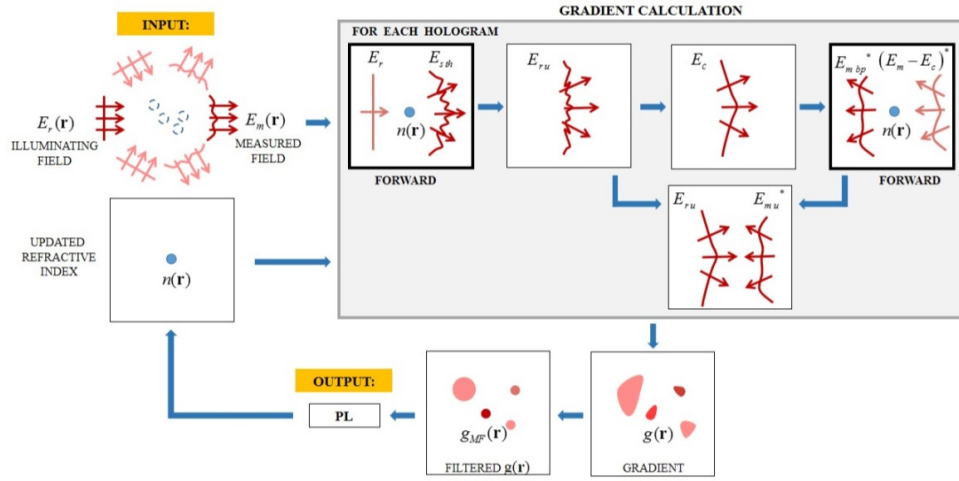
Fig. 2. Schematic process of NLODT-P model.

- *Updated illuminating field*: $E_{ru}^{i}(n, \mathbf{r})$. The presence of particles modifies the illuminating beam as the particles already located need to be taken into account. For this reason the forward solver computes the theoretical scattered field $E_{sth}^{i}(n, \mathbf{r})$ due to the available estimation of $n(\mathbf{r})$. The updated illuminating beam will be the sum of the original illuminating field and the scattered field $E_{ru}^{i}(n, \mathbf{r}) = E_{r}^{i}(\mathbf{r}) + E_{sth}^{i}(n, \mathbf{r})$.

- *Updated measured field*: $E_{mu}^{i}(n, \mathbf{r})$. The measured field can be explained partially by the particles already located. In fact, the expected measured field, for a given $n(\mathbf{r})$, can be computed as $E_{c}^{i}(n, \mathbf{r})$ by applying a frequency filter to the updated illuminating field $E_{ru}^{i}(n, \mathbf{r})$ in order to remove the frequencies which cannot be measured at far field or by the NA of the optical system. The difference between this computed measured field and the original measured field $(E_{m}^{i}(\mathbf{r})-E_{c}^{i}(n, \mathbf{r}))$ is the field that still need to be explained. This difference field is back-propagated through the available estimation of $n(\mathbf{r})$ and subsequently the updated measured field is computed. Let us remark that the conjugated difference field: $(E_{m}^{i}(\mathbf{r}) -E_{c}^{i}(n, \mathbf{r}))^{*}$ will play the role of the illuminating beam of the second call to the forward solver.

As in the initial iteration, a separate contribution to the gradient $g(\mathbf{r})$ should be obtained for each hologram. The gradient $g(\mathbf{r})$ of the cost function provides the distribution we should add to the estimated scattering potential $f(\mathbf{r})$ to minimize the cost function according to the classical Conjugated Gradient Method. This gradient can be taken as an image of only the

particles that remain to be located. However, our model does not need to solve the full image problem, just to find a new particle position.

Step 4. Locate next particle and exit

As for the initial iteration, the most probable position of next particle is the absolute maximum of the matched-filtered gradient $g_{MF}(\mathbf{r})$. The position is stored in the output variable *PL*, and the peak value, *value*, is used to decide the end of the model. Experience shows that *value* will decrease in each iteration and a convenient threshold ensures the process stops. The criterion to select an appropriate threshold value is not addressed in this work. Thus, a maximum number of iterations is also imposed.

Although the gradient can be expressed by an analytical equation, it still requires the resolution of the Helmholtz equation (*Forward* procedure) twice by each holographic recording. The Helmholtz equation is an example of a linear elliptic Partial Differential Equation (PDE), which has been extensively studied [16]. It can be numerically solved by means of an appropriate transformation based on Green's functions and a spatial discretization [16], for example, Finite Element Method (FEM) [18]. FEM discretizes the region of interest in small elements, assuming the function $E(\mathbf{r})$ can be approximated to a constant value in each of these elements. A regular mesh of elements is usually considered when the object shape is the unknown (inverse problems). So, the spatial derivatives of the Laplace operator can be discretized with a seven-point stencil in 3D. Thus, when a discretization process based on FEM and a spatial regular 3D mesh is applied to Eq. (1), the resultant linear system of equations is defined by $Mx = b$, where the independent term, b, depends on the illumination field, the unknown vector, x, identifies the scattered field and the matrix M, is related to the refractive index. This matrix is sparse and exhibits a strong regularity in both the pattern and the values of its non-zero elements, because it only includes seven non-zero diagonals. Let us note that a similar matrix will be obtained with a Finite Difference Method (FDM) approach with a second order approximation and the same regular mesh. The matrix M is very large and its size depends on the number of spatial discretization points or voxels into the volume (*Vol*) [14]. To avoid instability effects the voxels should be located such that the nearest neighbors are not further away than one tenth of the wavelength [19]. So, the *Vol* value results very high. Therefore, the resolution of a large size linear system of equations composed by complex numbers is included in *Forward* procedures. This consumes most of the computational resources of NLODT-P.

3. GPU-Based implementation of NLODT-P

NLODT-P model demands very high memory requirements and consumes long runtime, since every iteration includes the resolution of two Helmholtz equations. So, for the NLODT-P implementation, it is essential to apply: (i) specific approaches to reduce its memory requirements and (ii) High Performance Computing techniques to decrease its runtime. We have developed an approach to accelerate the resolution of the Helmholtz equation based on the exploitation of the regularities of the coefficient matrix (previously denoted as *M*) and the use of GPU computing.

The most computationally demanding procedure of NLODT-P is *Forward* (95% of the total runtime of the NLODT-P model), which is devoted to the resolution of the Helmholtz differential equation. We have considered the Biconjugate Gradient Method (BCG) for solving the large systems obtained from the discretization of the Helmholtz equations [20]. It iteratively computes two sparse matrix vector operations (SpMVs) that consume most BCG runtime [21]. BCG is a non-stationary iterative method to solve linear systems of equations $Mx = b$, where the matrix $M \in C^{n \times n}$ can be non-symmetric. Figure 3 shows the BCG method in a schematic way.

The key operations in the BCG method are two kinds of vector operations (dot and saxpy) and two sparse matrix vector products (SpMVs), computed at every iteration. It is remarkable that the symmetry of the Helmholtz equation has allowed the use of M instead of M^T.

Require: Define ϵ = Accuracy Threshold
Ensure: The value of $x^{(i)}$

1: Compute $r^{(0)} = b - Mx^{(0)}$ for some initial guess $x^{(0)}$
2: Choose $r'^{(0)} = r^{(0)}$; $p'^{(0)} = 0$; $p^{(0)} = p'^{(0)}$ $\rho'^{(0)} = 1$
3: Calculate $\Delta^{(0)} = \text{norm } 2(r^{(0)})$
4: **for** $i = 1, 2, \ldots$ until convergence **do**
5: $\quad\rho^{(i)} = (r'^{(i-1)}, r^{(i-1)})$
6: $\quad\beta^{(i)} = \rho^{(i)}/\rho'^{(i-1)}$
7: $\quad p^{(i)} = r^{(i-1)} + \beta^{(i)} p^{(i-1)}$
8: $\quad p'^{(i)} = r'^{(i-1)} + \beta^{(i)} p'^{(i-1)}$
9: $\quad v^{(i)} = M p^{(i)}$
10: $\quad\alpha^{(i)} = \rho^{(i)}/(p'^{(i)}, v^{(i)})$
11: $\quad x^{(i)} = x^{(i-1)} + p^{(i)}\alpha^{(i)}$
12: $\quad r^{(i)} = r^{(i-1)} - v^{(i)}\alpha^{(i)}$
13: $\quad r'^{(i)} = r'^{(i-1)} - \alpha^{(i)}(M^T p'^{(i)})$
14: $\quad\Delta^{(i)} = \text{norm } 2(r^{(i)})$
15: $\quad$**if** $\Delta^{(i)} < \epsilon\Delta^{(0)}$ **then**
16: $\quad\quad$**return** $x^{(i)}$
17: $\quad$**else**
18: $\quad\quad\rho'^{(i)} = \rho^{(i)}$
19: $\quad$**end if**
20: **end for**

Fig. 3. Algorithm of the Biconjugate Gradient Method.

Bearing in mind that the coefficient matrix M, obtained from the discretization of 3D Helmholtz equation, exhibits several regularities, the performance of SpMVs involved in the BCG method can be improved when its regularity is taken into account. So, we have developed an approach to accelerate the resolution of the Helmholtz equation using the BCG solver by exploiting the regularities of matrix M and the GPU computing.

Regularities of M allow us to define it by a specific storage format, which stores the minimal information of the Helmholtz matrix. In this way, both, memory requirements and BCG runtime are considerably reduced since this compact format requires less memory accesses to read sparse matrix elements. Hereinafter the format which takes advantage of the regularities of M is referred to as Compressed Regular Format (CRF) [22].

The regularities of M are the following: (1) M is symmetric; (2) non-zero elements are located at seven diagonals in the matrix, where one is the main diagonal, two of them are the first lower and upper diagonals and four of them are located by $\pm$ D1-th and $\pm$ D2-th diagonals; (3) all the elements of every lateral diagonal are equal to 1.

From these characteristics, the CRF consists of: (1) One array, $M[]$ (complex) of dimension Vol; and (2) two additional integer values (D1, D2) which specify the displacements of the lateral diagonals with respect to the main diagonal.

CRF optimally minimizes the amount of data needed to store the sparse matrix. The computation time of SpMV (included in BCG) is decremented because: (1) the number of memory accesses to read the elements of the sparse matrix is reduced; and (2) the number of float operations is reduced because six complex products can be avoided to compute every element of the output vector (bearing in mind the known value, one, of the lateral diagonals elements). CRF significantly reduces the memory requirements (in a factor of ten) and the

arithmetic operations which are involved in SpMVs, with respect to the COO format (default in MATLAB). Table 1 shows the memory requirements (in GB) to store the sparse matrix M with several sizes, using the COO format (MATLAB) and CRF.

Table 1. Memory Requirements (in GB) to Store the Sparse Matrix M with Several Sizes, Using the COO Format (MATLAB) and CRF

Vol	COO (MATLAB)	CRF
200^3	1.25	0.12
220^3	1.66	0.16
240^3	2.16	0.21
260^3	2.75	0.26
280^3	3.43	0.33
300^3	4.22	0.40

Additionally, the specific BCG method based on CRF has been accelerated using the GPU computing because GPUs offer desktop massive parallelization that can strongly reduce the runtime of this kind of computations. CUDA is a parallel interface that enables to increase the performance by harnessing the power of GPUs [23]. The GPU version of BCG is based on CUBLAS library [24] for accelerating vector operations and two different kernels for SpMVs based on CRF.

Preliminary implementations of NLODT-P have been developed by means of MATLAB framework [7]. So, it is necessary to assemble both languages (MATLAB and CUDA) to have the ease of use of MATLAB and the acceleration of GPU computing. Specific MEX-files routines [15] combined with MATLAB have been developed for accelerating the most computational costly procedure of the model (*Forward* procedure). MEX-files are dynamically linked subroutines produced from C, C + + or CUDA source code that, when compiled, can be run from within MATLAB in the same way as MATLAB files or built-in functions.

In order to analyze the computational performance of NLODT-P, numerical experiments with several volume sizes have been developed. These experiments execute NLODT-P for the location of four particles of 0.5μm diameter in the volumes which range from 200^3 to 280^3 voxels from the input information of three holograms. For the evaluation, MATLAB R2012a (enable multithreading) has been executed on a CPU (2x4 Intel Xeon E5620 cores, 48 GB RAM, 2.4 GHz clock speed and under Linux). Moreover, a GPU device NVIDIA Tesla M2090 has been considered. The NVIDIA GPU is used to accelerate the *Forward* procedure.

Figure 4 shows the execution runtime (in seconds) of NLODT-P using both approaches: only MATLAB framework and CUDA-MATLAB combination. GPU implementation of NLODT-P reduces the runtime by a factor of 37 from the initial MATLAB version. In other words, we have obtained acceleration factors up to 37 × with respect to the approach that uses neither GPU computing nor the CRF format. Therefore, the use of MATLAB, GPU computing and an ad hoc format to store the regular sparse matrix makes feasible the extension of the model.
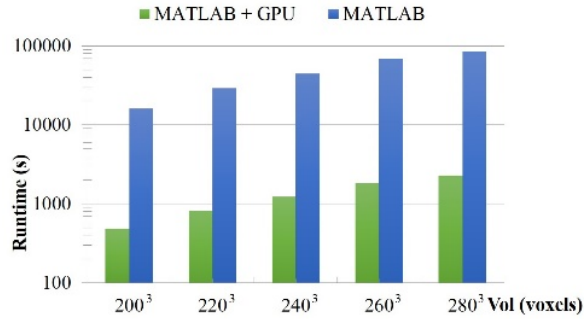
Fig. 4. Comparison of the NLODT-P runtime for MATLAB + GPU (GPU computing and CRF format) and the approach only with MATLAB.

4. Numerical experiments

The seeding concentration determines the maximum spatial resolution of the measured velocity field. Thus, NLODT that tackles the multiple scattering effects, could increase the classical particle velocimetry performance. Multiple scattering can be due to: (1) the particle finite size and (2) the close presence of other particles that modifies both, the illuminating beam and the scattered field diffracted by each particle. NLODT-P aims to resolve the particle location problem thanks to the matched-filtering strategy and the iterative procedure.

The relevance of the two sources on the particle image noise depends also on the optical configuration. Thus, two numerical experiments with different observation configuration have been studied: with full optical access [Fig. 5] and with only one observation direction [Fig. 8]. An apparently simple particle distribution consisting on four 2μm particles has been chosen. The illumination direction for each hologram is chosen along x, y and z axis respectively: they are illustrated in the figure by the slim red arrows [Fig. 5(a)]. The particles are stuck together and there are placed so that there is always one particle obscured by the others.
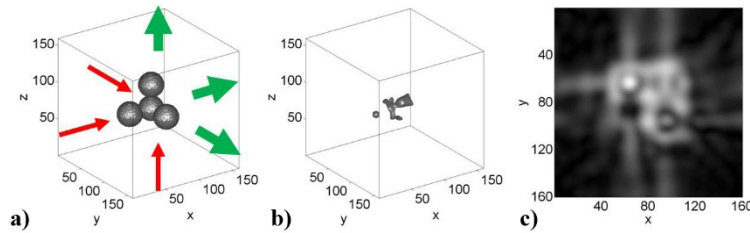

Fig. 5. (a) 3D view of the particle distribution problem with the full optical access recording configuration. LODT image: (b) 3D view of the brightest voxels and (c) 2D view of the plane at z = 64 pixels.

We consider a typical coherent illumination provided by a He-Ne laser, with λ = 0.633μm and that the holograms are recorded at far field with a NA = 0.55 microscope objective. The particle refractive index has been chosen as n = 1.33 (water) and the embedding media is considered air (n = 1.00). The volume of interest has been divided in $160 \times 160 \times 160$ voxels of one tenth of the wavelength. First, for simplicity, the observation directions have been chosen so they coincide with the illumination direction for each hologram: they are illustrated by the gross green arrows. This configuration maximizes the extension of the measured scattering spectrum by the three holograms, but it requires full optical access.

Figure 5(b) shows the 3D shape of the surface drawn by the 1‰ brightest voxels of the LODT image of the particle distribution problem. To better illustrate the blurred image, the modulus of the gradient at the plane $z = 64$ pixel that it is centered on the bottom three particles is also shown on Fig. 5(c). It is clear that the position of the particles cannot be

recovered from the LODT image. The relevance of the particle finite size to multiple scattering in this case is apparent when an isolated particle is considered. The LODT image of one particle recovered from similar three in-line holograms has been computed separately: the 3D view is shown on Fig. 6(a), and the modulus at the central plane on Fig. 6(b). The LODT image of the isolated particle is far from a spherical or even an Airy-like distribution. This complex amplitude has been referred as *sample* in the previous section since a similar pattern should be expected around each particle position on the LODT image for any other 2 μm-particle distribution. Thus, *sample* can be used to compute a matched-filter and to unravel the LODT image of Fig. 5(b)-5(c). Now, from the filtered LODT image, the four peak positions can be clearly identified [Fig. 6(c)-6(d)].
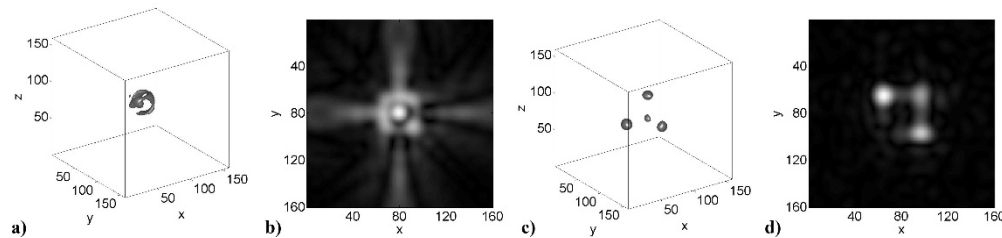
Fig. 6. LODT image of one isolated particle (*sample*): (a) 3D view and (b) 2D view of the central plane (at z = 81 pixels). Filtered LODT image of the particle distribution problem: (c) 3D view and (d) 2D view of the plane at z = 64 pixels.

As expected, NLODT-P can also solve this particle distribution problem. At each iteration, *sample* is also used to compute the filtered gradient of the cost function. The brightest peak of the filtered gradient determines the particle located in each iteration. Figure 7 shows from (left to right) the filtered gradient of the first iteration (which coincides with the filtered LODT image) and of the following iterations. For this first experiment, raw LODT does not resolve the particle distribution; meanwhile filtered LODT and NLODT do resolve the problem with similar location errors, around 2.1 pixels (0.13μm). Thus, the use of a matched-filter, using a-priori information of the particles, was enough to obtain the particle positions.
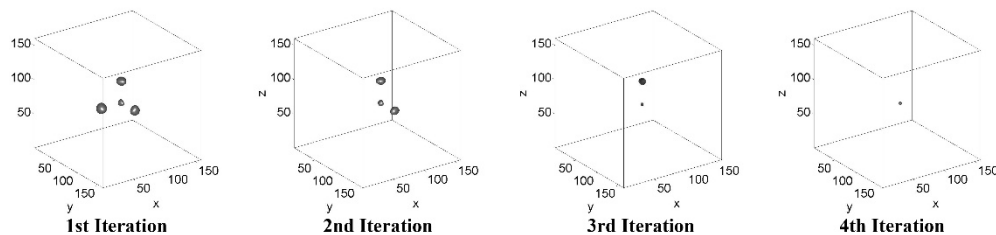

Fig. 7. Filtered NLODT gradients (images of the particles that remain to be located) for the full optical access configuration.

Ideally, for an example with experimental data we would expect around one thousand particles. If the matched-filter could unravel several particles at once, a straightforward modification in the line 14 of the algorithm [Fig. 1] would allow recovering the position of several particles at every iteration. However, as the next simulated experiment shows, the maximum number of particles, which can be located at every iteration, is a parameter that has to be studied carefully, as it depends not only on the particle concentration but also on the optical recording system.

On a second experiment we have considered a more realistic set-up: the illumination directions are from three orthogonal directions as before, but now all the holograms are recorded from the same observation direction. This simplifies the calibration of the recording

set-up and the subsequent superposition of the image obtained from each hologram. It is clear, that the spatial resolution will be now reduced and particle images will be elongated along the observation direction. This observation direction has been chosen as shown in the Fig. 8(a) so they form an angle of 45° with any of the illumination direction. This means that the illuminating field and the forward scattering cannot be recorded by air-immersion objectives, and subsequently low frequencies of the object are not recorded [8].
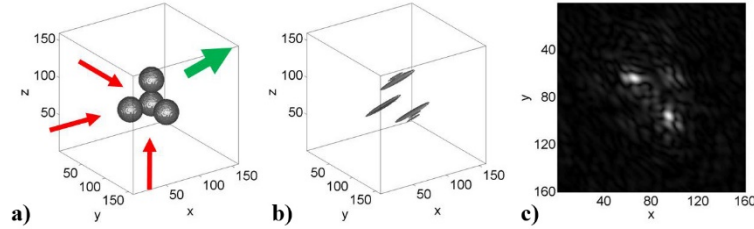
Fig. 8. (a) 3D view of the configuration with one observation direction. Filtered LODT image: (b) 3D view of the brightest voxels and (c) 2D view at z = 64 pixels.

The LODT image does not resolve the particle distribution with this observation configuration. Furthermore, even after computing the corresponding *sample* and the filtered LODT image [Fig. 8(b) and 8(c)], only three particles can be located. It is relevant to note that if we admit these three peaks as valid particle positions in this first iteration, next gradient would be a noisy distribution without a clear peak. Thus, the algorithm would finish here or its output would include some new wrong particle locations, depending on the finishing criterion. In this experiment, the high particle concentration and the limited observation directions make the inverse problem very sensitive to the initial particle locations. Thus, only one particle location per iteration has been admitted.

The filtered gradient obtained at every iteration of NLODT-P model is shown in Fig. 9.
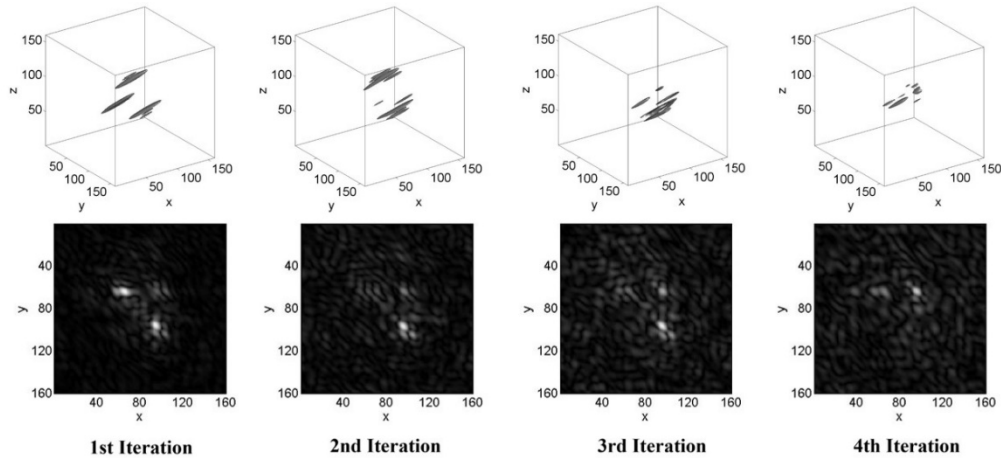

Fig. 9. Filtered NLODT gradient for the configuration with only one observation direction: (top) 3D view and (bottom) 2D view at z = 64 pixels.

From the figure, we can appreciate how the visibility of the particle at the corner (x,y,z) = (96,64,64) increases with each iteration. The final position error is roughly 0.3μm, even though, the performance of NLODT-P is clearly advantageous compared to the linear approach as it is apparent from these small numerical experiments. Additionally, the combination of MATLAB and GPU for solving the *Forward* problem, previously discussed, makes feasible the study of such kind problems and, subsequently, the use of NLODT-P for fluid velocimetry applications.

5. Conclusions and future works

In this paper a three dimensional non-linear ODT model to locate particles has been presented, as part of a fluid velocimetry technique. Two numerical experiments, with different observation configuration but identical illumination set-up, have been considered. They illustrate that multiple scattering influences the object visualization in a different way depending on the recording configuration. These experiments also demonstrate the advantages of NLODT-P vs LODT performance. It has been shown that multiple scattering due to the particle finite size can be tackled with a suitable matched filter, but multiple scattering between particles have to be addressed with an iterative optimization method such as NLODT-P.

We have implemented the NLODT-P model using MATLAB combined with GPU computing by means of MEX-files due to the fact that MEX-files allow the combination of CUDA-GPU programming and MATLAB. Acceleration factors range up to 37 × with respect to the approach that only uses MATLAB framework. Additionally, the use of a specific format to store the large sparse matrix (*M*), involved in the BCG method, has reduced the memory requirements by a factor of ten with respect to the traditional format for specifying sparse matrices in MATLAB. In this line, we are currently working on the implementation of a hybrid code of the *Forward* procedure, based on the combination of CPUs and GPUs in order to take advantage of the heterogeneity of the resources of the modern high performance architectures [25]. This hybrid implementation could allow us to extend the resolution of this kind of problems.

As consequence of this work we can say the NLODT-P is a good approach to improve the accuracy of linear ODT methods in locating seeding particles in fluid velocimetry applications. High Performance Computing platforms and tools are essential to develop and apply this approach. However, further work is needed towards a multiscale model for embedding these proposals in production environments.

Acknowledgment