

Intercorrélation d'images PIV sur GPU – application à un écoulement en fluide turbide

Emmanuel CID¹, Omar Bentata et Dominique Anne-Archard

*IMFT, Université de Toulouse (INPT, UPS) et CNRS, Allée Camille Soula, Toulouse
F-31400, France*

1 Introduction

La Vélocimétrie par Images de Particules (PIV) est une technique de mesure largement utilisée en Mécanique des Fluides, faisant intervenir une source Laser associée à une ou plusieurs caméras. L'utilisation de capteurs matriciels de haute résolution (\geq à 4 Mpixels) se généralise aussi bien pour des systèmes d'acquisition basse cadence (de l'ordre de quelques Hertz), que pour des systèmes rapides autorisant des fréquences d'acquisition de plusieurs KiloHertz. Il devient désormais usuel de générer des centaines de GigaOctets de données lors d'une étude expérimentale. Ces données sont ensuite traitées numériquement avec des algorithmes itératifs de traitement d'images pour lesquels on recherche une précision et une robustesse suffisantes. Avec des algorithmes exécutés de manière séquentielle, ce temps de traitement est largement supérieur au temps nécessaire pour l'acquisition des données.

Nous proposons, dans cette communication, de détailler l'algorithme d'intercorrélation d'images que nous avons porté sur une architecture GPU (Graphics Processing Units), particulièrement adaptée aux tâches de traitement d'images parallélisées. Cette application GPU, développée sous CUDA, a été utilisée pour étudier la propagation d'anneaux tourbillonnaires en fluide non newtonien par PIV2D2C.

2 Accélération du calcul d'intercorrélation d'images PIV sur GPU

La Vélocimétrie par Images de Particules (2D2C) permet de mesurer des champs de vitesse instantanées selon une coupe d'un écoulement. Un éclairage plan, assuré par une nappe laser, illumine une tranche d'un écoulement ensemençé de traceurs. Ces particules ou gouttelettes diffusent la lumière laser. Une caméra capture l'image des traceurs à deux instants successifs. Deux images I1 et I2 sont enregistrées à t et t+dt (figure1).

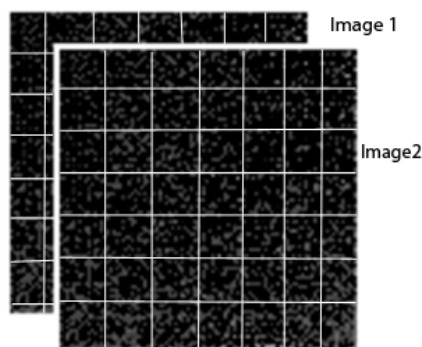


Figure 1 – Doublet d'images de particules découpées en mailles d'analyse

Les images sont découpées en mailles d'analyse ou régions d'interrogation. Pour chaque maille, la fonction de corrélation croisée centrée normalisée (ZNCC) est calculée et la détermination de la

¹ Correspondant : Emmanuel.Cid@ensiacet.fr

position de son maximum par rapport au centre de la maille d'analyse est une mesure du déplacement du groupe de particules pour cette région d'interrogation.

Soit $I1(x,y)$: la fonction représentant les niveaux de gris de l'image 1 de dimension $M_x \times M_y$ au point de coordonnées (x,y) , et $I2(x,y)$: la fonction correspondante pour l'image 2 de dimension $N_x \times N_y$ au point de coordonnées (x,y) ,

La fonction ZNCC en chaque point (u,v) s'écrit :

$$ZNCC(u, v) = \frac{\sum_{x,y} (I1(x, y) - \overline{I1_{u,v}})(I2(x - u, y - v) - \overline{I2})}{\sqrt{\sum_{x,y} (I1(x, y) - \overline{I1_{u,v}})^2 \sum_{x,y} (I2(x - u, y - v) - \overline{I2})^2}}$$

avec
$$\overline{I1_{u,v}} = \frac{1}{M_x M_y} \sum_{x=u}^{u+M_x-1} \sum_{y=v}^{v+M_y-1} I1(x, y)$$

et
$$\overline{I2} = \frac{1}{N_x N_y} \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} I2(x, y)$$

La fonction ZNCC peut être calculée directement selon l'équation ci-dessus. D'après le théorème de *Wiener-Khintchine*, elle peut également être implémentée après passage dans le domaine spectral par un produit de transformées de Fourier. Cette dernière méthode est moins coûteuse en nombre d'opérations [1]. De plus, les bibliothèques de transformées de Fourier rapide (FFTW par exemple) présentent des performances intéressantes, aussi bien en terme de rapidité, que de choix de dimension de maille d'analyse. Cette méthode par FFT est donc très répandue [2]. Les algorithmes PIV standard utilisent des processus itératifs afin d'améliorer la précision du calcul et sa robustesse. Certains auteurs [3], [4] ont développé des algorithmes PIV basés sur le calcul direct de la fonction d'intercorrélation.

Dans cette communication, nous allons détailler l'implémentation du calcul de la fonction ZNCC pour chaque maille d'analyse PIV sur une architecture GPU.

2.1 Architecture GPU et CUDA

CUDA (Compute Unified Device Architecture) est une technologie de GPGPU (*General-Purpose Computing on Graphics Processing Units*). Un processeur graphique (GPU) est utilisé pour exécuter des calculs généraux habituellement exécutés par le processeur central (CPU) de l'ordinateur. CUDA a été développé par NVIDIA pour programmer et exécuter des tâches sur les cartes graphiques de la même marque.

Un GPU performant est constitué de plusieurs centaines de processeurs (voir tableau 1). Ces processeurs sont les équivalents des coeurs pour les CPU, qui en comportent seulement quelques unités. La possibilité de parallélisation d'un calcul est donc supérieure avec un GPU par rapport à un CPU équipant une station de travail standard.

Si l'on considère une portion parallèle de code à exécuter sur le GPU, chacune de ses instances s'appelle un thread. Une grille est constituée de blocs. Chaque bloc regroupe des threads. (fig 2)

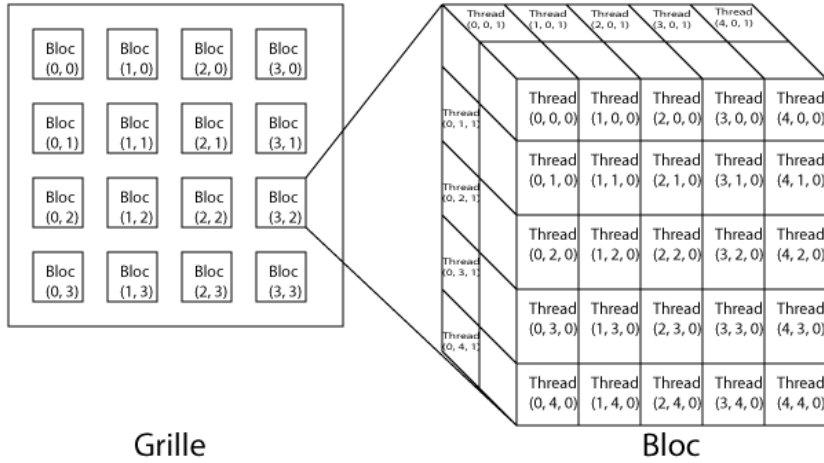


Figure 2 – Schéma de parallélisme avec CUDA

Cette parallélisation matricielle est particulièrement adaptée au traitement d'images si l'on affecte un thread à chaque pixel. Elle est particulièrement bien adaptée au traitement PIV si l'on identifie les blocs de threads aux mailles d'analyse à traiter indépendamment les unes des autres (voir figure 1 et 2).

2.2 Algorithme d'intercorrélation d'images PIV implémenté sous CUDA

Nous avons fait le choix d'implémenter l'algorithme du calcul direct de la fonction ZNCC sous CUDA. Pour des dimensions de maille d'analyse standard (16^2 pixels), nous avons comparé les deux implémentations sous CUDA (calcul direct et calcul par FFT). Le calcul direct sous GPU, pour cette dimension de maille, est apparu plus rapide à l'exécution.

Nous allons détailler ci-dessous l'algorithme du calcul direct de la fonction ZNCC mis en œuvre, pour une image découpée en $k \times l$ mailles d'analyse (sans recouvrement) :

1. Transfert des images entières $I1$ et $I2$ vers la mémoire GPU de type texture
2. Calcul du terme $\overline{I2_{k,l}}$ pour chacune des $k \times l$ mailles d'analyse :

$$\overline{I2_{k,l}} = \frac{1}{NxNy} \sum_{i=1}^{Nx} \sum_{j=1}^{Ny} I2(i + k * blockDim.x, j + l * blockDim.y)$$

blockDim fait référence aux dimensions d'un bloc de threads (équivalent à Nx et Ny). Ce calcul de sommation utilise l'algorithme de réduction [5]. Les valeurs $\overline{I2_{k,l}}$ sont stockées dans une mémoire globale.

3. Calcul du terme $\overline{I1_{u,v}}$ sur tous les pixels de l'image $I1$:

$$\overline{I1_{u,v}} = \frac{1}{MxMy} \sum_{o=u-My/2}^{u+My/2} \sum_{p=v-Mx/2}^{v+Mx/2} I1(o, p)$$

Pour ce terme-ci, la méthode de sommation est similaire à celle utilisée dans [6]. Des threads exécutent des sommes de niveau de gris sur chaque colonne de dimension My . Cette 1^{ère} somme est conservée en mémoire afin que d'autres threads effectuent une somme glissante de dimension Mx centrée sur chaque colonne p (figure 3).

Les $\overline{I1}_{u,v}$ sont stockés dans une texture en mémoire GPU.

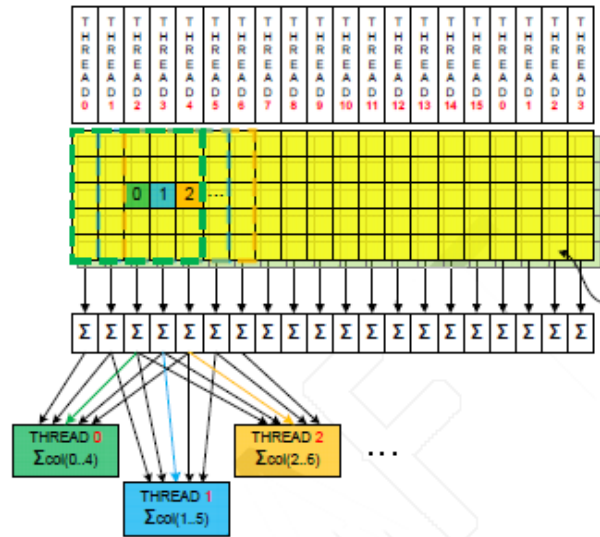


Figure 3 – Schéma de calcul du terme $\overline{I1}_{u,v}$

4. La parallélisation du produit de corrélation est organisée suivant un nombre de threads identique au nombre de pixels de l'image I2 et regroupés par blocs de dimension $N_x \times N_y$. Pour optimiser les temps d'accès mémoire, on charge en mémoire partagée les niveaux de gris de la maille pour I2 soustraits de $\overline{I2}_{k,l}$ (calculé à l'étape 2). On effectue la même opération pour la maille de l'image I1 soustraite des $\overline{I1}_{u,v}$ (calculé à l'étape 3). Le produit de corrélation est effectué dans une boucle où l'on somme le numérateur de la fonction d'intercorrélation et les deux membres du dénominateur. Puis on effectue la normalisation de la fonction d'intercorrélation avec ces trois membres.
5. La recherche de la position du maximum de la fonction de corrélation est parallélisée pour chaque maille d'analyse. Deux interpolations paraboliques sont effectuées successivement sur les deux dimensions de la maille d'analyse afin de déterminer une position en valeur sub-pixellique. Le rapport entre les deux pics de plus grandes amplitudes (RSB) est également calculé dans ce kernel.
6. Enfin, le transfert des résultats $(U, V, RSB)_{k,l}$ pour chaque maille d'analyse est effectué de la mémoire globale du GPU vers le CPU.

En pratique, cet algorithme est répété itérativement des étapes 2 à 5 en décalant symétriquement, la position des mailles d'analyse sur les images [7]. Le décalage des mailles étant réalisé en valeur décimale, l'interpolation des imagettes est de type B-spline cubique [8].

A l'étape 4, nous avons évoqué l'utilisation de mémoire GPU partagée pour le calcul de la fonction d'intercorrélation. Ce type de mémoire est directement dépendant des caractéristiques des cartes graphiques considérées. Pour les cartes NVIDIA de capacité de calcul 1.3, cette utilisation n'est possible que pour des mailles de taille 16^2 pixels ou inférieur. Afin d'éviter cette limitation, nous nous sommes appuyés sur une décomposition du calcul de la fonction d'intercorrélation telle qu'utilisée par Rohály et al. [9].

2.3 Temps de calcul sur GPU

Les temps de calcul, pour un couple d'images 2048x2048 pixels, sont quantifiés avec différentes cartes graphiques, incluant les temps de transfert mémoire CPU↔ GPU. Les temps d'exécution en fonction du nombre de cœurs équipant les cartes sont donnés ci-dessous (tableau1). Le processus complet fait intervenir quatre itérations sur l'ensemble de l'image : 1 itération avec un maillage original en 32² pixels suivie de 3 itérations avec un maillage final de 16² pixels sans recouvrement.

carte graphique NVIDIA	nombre de cœurs	temps d'exécution en seconde
GeForce GTX9800	128	3,5
Tesla C1060	240	1,8
Quadro 5000	352	1,0

Tableau 3 – Temps de calcul d'un champ de vecteur pour une image 2048*2048 et un maillage final de 16² pixels, calculé par ZNCC pour différentes cartes graphiques.

D'autres algorithmes dédiés au calcul PIV ont déjà été portés sur GPU. Tarashima et al. [10] ont implémenté un algorithme similaire à celui proposé dans cette communication mais sans détailler l'algorithme mis en œuvre et en utilisant une interpolation bilinéaire. Champagnat et al. [11] ont proposé une méthode de flot optique très performante en temps de calcul car basée sur des opérations pixel à pixel et une convolution séparable : plus rapide que la convolution non séparable mise en œuvre ici. Cependant, la fonction ZNCC reste un outil standard pour le calcul PIV en terme de robustesse par rapport à une large gamme d'écoulement [1]. De plus, compte-tenu du faible temps d'exécution de notre algorithme (1,0 seconde pour une image 2048² avec calcul ZNCC itératif sur un maillage final de 16² pixels), le principal gain (x20) est réalisé par rapport au calcul PIV spectral parallélisé sur CPU.

3 Application à un cas expérimental

Les résultats expérimentaux sur lesquels nous proposons des tests comparatifs sont issus de l'étude du développement d'un anneau tourbillonnaire en fluide non newtonien. Le dispositif expérimental est présenté sur la figure 4. L'anneau est généré par un système cylindre-piston (ϕ 21mm) dans une cuve de section carrée 240x240 mm². Le cylindre est centré dans la cuve et on réalise des mesures du champ de vitesse dans un plan diamétral sur une fenêtre 80x80mm². Une visualisation par fluorescéine est proposée sur la figure 5 (gauche). La chaîne PIV est constituée d'un laser Quantel Nd-YAG CFR200 et d'une caméra PCO 2000 2048x2048 pixels. L'ensemencement est assuré avec des particules d'Orgasol, de diamètre $D_p=60 \mu\text{m}$.

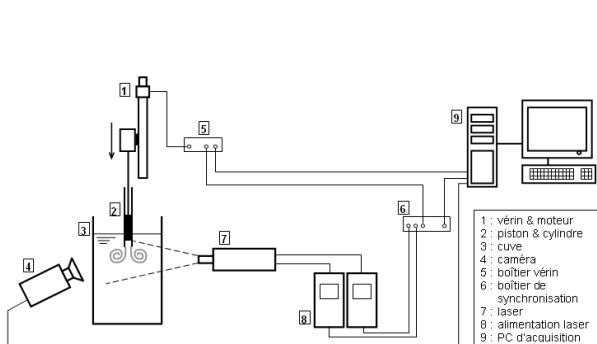


Figure 4 - Montage expérimental

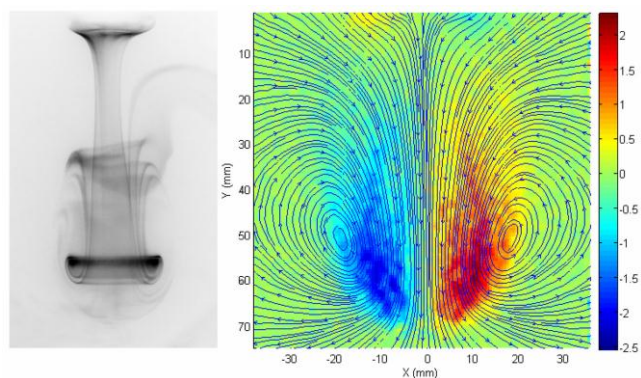


Figure 5 - Anneau tourbillonnaire en fluide non newtonien (solution de Xanthane) : visualisation, lignes de courant et champs de vorticité ($Re=90$)

L'une des difficultés de ce travail expérimental provient de la turbidité des fluides utilisés (solutions de Xanthane pour obtenir un comportement non newtonien rhéofluidifiant). Celle-ci entraîne un rapport « signal sur bruit » des images de particules qui est très pénalisant lors du traitement PIV.

Sur la figure 6, on représente des images PIV pour deux fluides ayant des turbidités différentes. Le cas de référence étant de l'eau à comparer avec une solution de Xanthane à 0.04% que l'on nommera X4. Dans les mêmes conditions d'observation, la moyenne, l'écart type et la valeur minimale des niveaux de gris représentent des critères intéressants pour rendre compte de la turbidité observée. Le tableau 4 résume des valeurs élevées pour la moyenne et la valeur minimale des niveaux de gris avec la solution de Xanthane par rapport à l'eau alors que la valeur d'écart-type est plus faible.

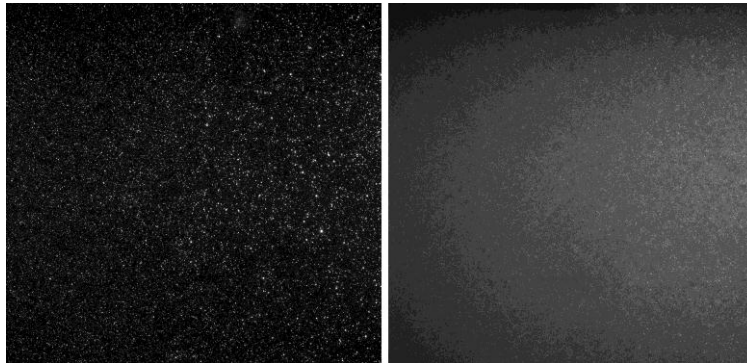


Figure 6 – à gauche : images brutes de particules d'Orgasol dans de l'eau. A droite : dans une solution de Xanthane X4

Valeur des Niveaux de gris	Moyenne	Ecart-type	Minimum	Maximum
eau	190	100	0	10000
Solution Xanthane X4	2600	70	1300	10000

Tableau 4 – Valeurs moyennes, écart-type, minimale et maximale de niveau de gris pour les images brutes de particules avec la solution de Xanthane et de l'eau.

La seconde des difficultés est liée à l'information recherchée dans cette étude, la vorticité, qui est calculée à partir des gradients de vitesse et qui nécessite donc des champs de vitesse de bonne qualité malgré une dynamique des vitesses relativement conséquente entre le centre du jet et les bords du champs de vitesse. Compte tenu du rapport signal à bruit défavorable de nos images, le calcul PIV itératif a été réalisé sur des mailles 32^2 avec un recouvrement de 50 %.

Les résultats (présentés sur la figure 5) montrent que l'obtention des dérivées spatiales de la vitesse, particulièrement sensibles au bruit de mesure, reste accessible au moyen de la fonction de corrélation normalisée (ZNCC) calculée dans notre implémentation sur GPU.

Sur la figure 7, on montre les champs instantanés de vorticité au même instant adimensionné ($t^*=4,6$) pour les deux fluides considérés calculés par intercorrélations sur GPU. On observe l'effet de la turbidité qui augmente sensiblement le bruit sur le calcul de la vorticité.

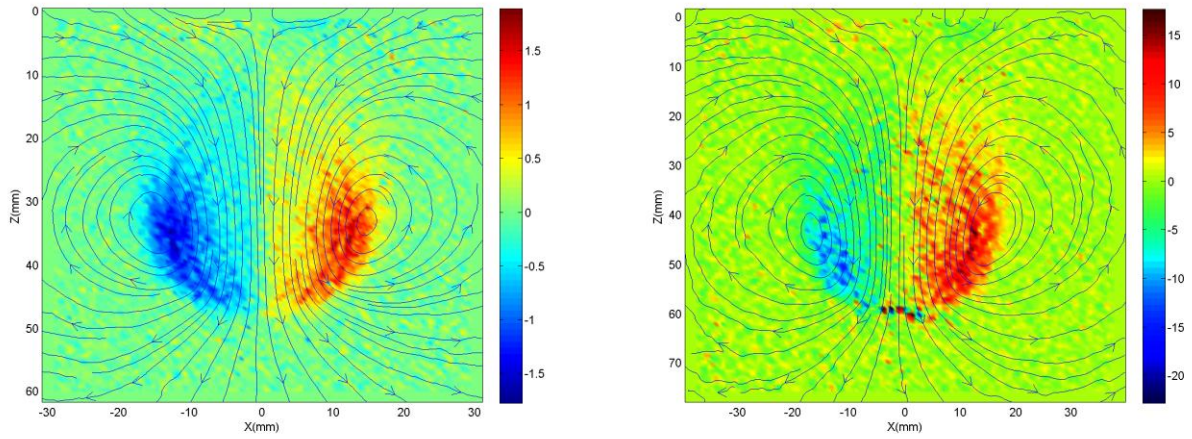


Figure 7 – Champs de vorticité instantanés à $t^*=4.6$ calculé par intercorrélation sur GPU. (à gauche pour l’anneau tourbillonnaire se développant dans de l’eau, à droite dans une solution de Xanthane X4)

Sur la figure 8, on compare les profils de vitesse instantanés, à $Z=44$ mm pour $t^*=4.6$ et la solution de Xanthane X4, calculés par intercorrélation sur GPU (corrélation directe) et sur CPU (corrélation spectrale). Les résultats sont similaires pour un temps de calcul diminué par 20 lors de l’exécution sur GPU.

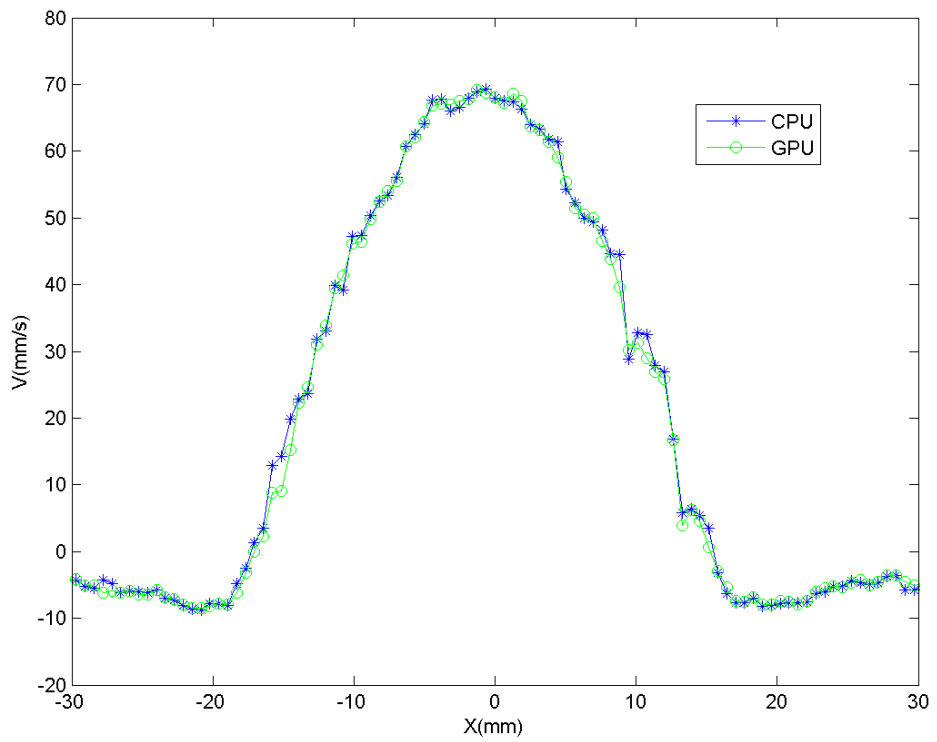


Figure 8 – Profils de vitesse instantanés calculés par intercorrélation sur GPU et sur CPU pour la même image de particules avec une solution de Xanthane X4 à $Z=44$ mm (selon la figure 7 droite)

4 Conclusion

Dans cette communication, nous avons détaillé un algorithme itératif, multi-grille, de calcul direct d'intercorrélation d'images PIV porté sur GPU. Les différentes étapes de l'implémentation sont documentées afin de partager l'expérience de cet outil en cours d'évolution. Le logiciel a été confronté à un écoulement délicat à traiter : le développement d'anneaux tourbillonnaires en fluide turbide. Les résultats obtenus avec la version GPU ont été comparés avec un calcul d'intercorrélation effectué par FFT sur CPU pour le même cas expérimental. L'intérêt du calcul GPU est particulièrement net en terme de rapidité de calcul pour des images de grandes dimensions, sans pénalisation sur la précision de mesure pour le cas expérimental étudié.

Par ailleurs, ce type de traitement peut être effectué en local, sans transfert de grandes quantités de données ; ce qui autorise un retour rapide sur la qualité des résultats mais également sur la validation du nombre de champs de vitesse nécessaire à la convergence des données statistiques à extraire.

Références

- [1] O.Pust, " PIV: Direct Cross-Correlation compared with FFT-based Cross-Correlation". In: Proceedings of the 10th International Symposium on Applications of Laser Techniques to Fluid Mechanics. Lisbon, Portugal, (2000)
- [2] M. Stanislas, K. Okamoto, C. J. Kaehler, J. Westerweel and F. Scarano, "Main results of the third international PIV Challenge", Experiments in Fluids, Vol.45, No.1, 27-71, (2008)
- [3] A. Fincham and G. Delerce, "Advanced optimization of correlation imaging velocimetry algorithms," Experiments in Fluids, Vol. 29, S013-S022, (2000)
- [4] A.Susset, JM. Most, D. Honore, "A novel architecture for a superresolution PIV algorithm developed for the improvement of the resolution of large velocity gradient measurements", ", Experiments in Fluids, Vol 40:70–79, (2006)
- [5] Harris, M.: Optimizing Parallel Reduction in CUDA. NVIDIA Developer Technology (2008)
- [6] J.Stam, "Stereo imaging with CUDA", NVIDIA technical report (2008) disponible à <http://openvidia.sourceforge.net/index.php/OpenVIDIA>.
- [7] B. Lecordier, M. Trinite, « Advanced PIV algorithms with image distortion validation and comparison using synthetic images of turbulent flow », in: Proceedings of the EUROPIV 2 workshop - Zaragoza, (2003)
- [8] D. Ruijters, B. Romeny and P. Suetens, "Efficient GPU-Based Texture Interpolation using Uniform B-Splines.", Journal of graphics, gpu and game tools – Vol. 13.4, Pages 61-69, (2009)
- [9] J. Rohály, F. Frigerio, D.P. Hart, "Reverse hierarchical PIV processing", Measurement Science and Technology, 13 (7), pp. 984–996 (2002)
- [10] S. Tarashima, M. Tange, S. Someya et K. Okamoto, « GPU accelerated direct cross-correlation PIV with window deformation », in: Proceedings of 15th International Symposium on Applications of Laser Techniques to Fluid Mechanics, Lisbon, Portugal, (2010)
- [11] F. Champagnat, A. Plyer, G. Le Besnerais, B. Leclaire, S. Davoust et Y. Le Sant, « Fast and accurate PIV computation using highly parallel iterative correlation maximization», Experiments in Fluids – 50, 1169–1182 (2011)