

Utilisation de la corrélation d'images en temps réel pour le pilotage d'un essai mécanique

V. Couty¹, J-F. Witz¹, J. Berthe², P. Lecomte¹, E. Deletombe², M. Brieu³

¹ LaMcube, Ecole Centrale de Lille, {victor.couty, jean-francois.witz, pauline.lecomte}@centralelille.fr

² ONERA, {julien.berthe, eric.deletombe}@onera.fr

³ Department of Mechanical Engineering, California State University-Los Angeles, mbrieu@calstatela.edu

Résumé — La corrélation d'images est un outil performant pour la mesure de champs mais qui nécessite un temps de calcul important, surtout avec l'augmentation de la résolution des caméras. La corrélation est donc généralement utilisée en post-traitement des essais. Pourtant, faire entrer des grandeurs obtenues par corrélation d'images dans le pilotage peut permettre des essais plus riches permettant de mieux cibler les phénomènes étudiés. Nous proposons ici une méthode de corrélation d'images applicable en temps réel et nous étudierons les applications d'une telle méthode, en particulier couplée avec la POD pour les champs cinématiques.

Mots clés — corrélation d'images, POD, mesure de champs, vision par ordinateur, GPU

1 La corrélation d'images

1.1 Définitions

La corrélation d'images numériques (CIN) est une technique de mesure de champs cinématiques par analyse d'images. Elle permet d'obtenir une mesure des déplacements d'un ensemble de points d'une surface. Cette technique est notamment utilisée en mécanique pour mesurer des déplacements locaux et par dérivation spatiale, des champs de déformation. Il s'agit d'une mesure non intrusive. La CIN consiste en la résolution d'un problème de minimisation permettant de trouver la transformation qui minimise les différences entre deux images prises au cours d'un processus d'évolution du système observé. Ce problème peut être formulé selon deux approches :

- L'approche locale : la corrélation est effectuée sur des sous parties de l'image et les résultats sont combinés pour obtenir le champ de déplacement sur l'image entière.
- L'approche globale : la corrélation traite l'intégralité de l'image, cela permet des champs denses au pixel, mais les discontinuités peuvent être plus difficiles à intégrer.

La CIN est utilisée depuis plusieurs décennies mais possède une contrainte majeure : le traitement des images est coûteux en calcul et sauf à résolution ou fréquence faible, le traitement prend généralement un temps non négligeable. Ce point oblige le plus souvent à effectuer les calculs de corrélation à posteriori. Les champs sont alors inaccessibles au cours de l'essai lui-même et ne peuvent donc entrer dans la commande de l'essai. Le traitement de la corrélation en temps réel est accessible depuis quelques années grâce à l'évolution du matériel informatique. Cette nouvelle approche pourrait permettre de s'appuyer sur des variables plus riches pour le pilotage d'un essai sur structure. Il est par exemple possible de piloter le maximum de déformation même si son emplacement n'est pas connu a priori.

Nous proposons ici une méthode permettant d'effectuer de la corrélation d'images globale. En particulier, une implémentation open source de cette méthode en Python et CUDA sera étudiée afin de discuter des performances de cette approche sur une machine grand public et de son intérêt au sein d'essais mécaniques.

1.2 Principe

Le problème de minimisation de la corrélation d'images s'exprime de la façon suivante : soient F et G deux fonctions $\mathbb{R}^2 \rightarrow \mathbb{R}$. L'objectif de la corrélation d'images est de trouver la fonction $u : \mathbb{R}^2 \rightarrow \mathbb{R}$ telle que $F(x)$ soit égal à $G(x + u(x))$ en tout point de l'image. Différentes contraintes seront imposées

concernant la forme de u , rendant impossible l'égalité stricte. La résolution du problème consistera alors à minimiser la différence entre les deux images. Considérons la définition suivante :

$$r = \int_{img} (F(x) - G(x + u(x)))^2 dx \quad (1)$$

L'équation (1) définit la *résidu* selon la méthode des moindres carrés. Il s'agit d'un scalaire qui quantifie l'écart entre l'image originale $F(x)$ et la seconde image "recalée" $G(x + u(x))$. À noter qu'il est possible d'utiliser d'autres métriques pour quantifier cet écart. La norme L2 est choisie ici entre autre pour son coût de calcul faible.

Les images numériques ne sont pas des fonctions continues mais des données discrètes. L'intégration devient donc une somme sur les pixels de l'image. Par ailleurs, l'ajout d'un masque permettra d'exclure les zones de l'image qui ne font pas partie du corps étudié. Le masque M est une fonction qui associe une valeur faible ou nulle dans les zones où la différence entre l'image n'est pas pénalisée et une valeur non nulle ailleurs. Son impact ne sera pas étudié dans la suite du raisonnement. L'expression prend alors la forme suivante :

$$r = \sum_{p \in pixels} M(p) \cdot (F(p) - G(p + u(p)))^2 \quad (2)$$

Il convient de noter que l'obtention d'une précision inférieure au pixel nécessite la possibilité d'évaluer l'image à des coordonnées non entières. Pour cela différentes méthodes d'interpolation peuvent être utilisées comme l'interpolation bilinéaire, bicubique ou B-spline [3].

1.3 Méthode pour la corrélation en temps réel

La méthode décrite ici se base sur une approche globale : l'image sera traitée dans son intégralité. Afin de limiter la dimension de l'espace de recherche de u , il sera recherché sous la forme d'une somme de champs choisis au préalable [5]. Le champs obtenu sera donc une projection du champs réel dans la base de champs choisie. Soient U_i les N champs de base choisis, $u(x)$ est donc défini comme suit :

$$u(x) = \sum_{i=1 \rightarrow N} k_i \cdot U_i(x) \quad (3)$$

Le choix de la base de champs $\{U\}$ va conditionner le résultat de la corrélation. Plusieurs approches sont possibles quant au choix de cette base : il est possible de choisir comme base vectorielle des champs avec un sens physique, une décomposition de champs expérimentaux ou encore extraits de simulations. Le choix de la base de champs sera discuté en partie 2.1.

Soit N le nombre de vecteurs qui composent la base $\{U\}$, la corrélation revient alors à chercher le vecteur $K = (k_i)$ de dimension N qui vérifie l'équation (3). Le vecteur K sera calculé de façon itérative, S étant le vecteur direction de recherche. À chaque itération, S sera ajouté à K , jusqu'à la vérification d'un critère d'arrêt basé sur la valeur du résidu ou le nombre d'itérations.

$$K_{r+1} = K_r + S_r \quad (4)$$

La méthode dite "Quasi-Newton" sera utilisée pour déterminer S [11]. L'hypothèse principale de la corrélation est la conservation du flot optique qui implique $F(x) \simeq G(x + u(x))$. Le développement limité à l'ordre 1 de G est écrit dans l'équation (5).

$$r = \int_x (F(x) - G(x) - u(x) \cdot \nabla G(x))^2 dx \quad (5)$$

En supposant que le déplacement u est petit, les termes en u^2 sont négligeables. En posant $D = F - G$ et en remplaçant l'intégrale par une somme, l'équation devient :

$$r = \sum_{p \in pixels} D(p)^2 - 2u(p) \cdot D(p) \cdot \nabla G(p) \quad (6)$$

Afin de gagner en temps de calcul, les gradients de F seront utilisés à la place des gradients de G . En effet, à la convergence de l'algorithme, $G(x + u(x)) \simeq F(x)$. Cette substitution permet de ne calculer que

les gradients de l'image d'origine et de les utiliser pour toute autre image G . Chaque composante k_i du vecteur K vérifiera alors l'équation (7).

$$\frac{\delta r}{\delta k_i} = \sum_{p \in \text{pixels}} \frac{\delta (D(x) \cdot k_i \cdot U_i(x) \cdot \nabla F(x))}{\delta k_i} = \sum_{p \in \text{pixels}} D(x) \cdot U_i(x) \cdot \nabla F(x) \quad (7)$$

Les termes $U_i(x) \cdot \nabla F(x)$ pourront être calculés au préalable et n'ont pas besoin d'être évalués pour chaque images. Il reste à estimer et inverser la matrice hessienne H . Celle-ci est estimée en se basant sur les champs et la première image puis sera réutilisée par la suite.

Les étapes de l'algorithme sont alors les suivantes :

- Lire U
- Lire F
- Calculer $\nabla F, H$ et H^{-1}
- Initialiser K_0
- Pour G
 - Calculer $D(x) = F(x) - G(x + u(x))$ et $r = \sum_x D(x)$
 - Si condition finale atteinte, fin de la boucle
 - calculer S
 - calculer $K_{i+1} = K_i + H^{-1} \cdot S$

La condition finale peut être une limite sur le nombre d'itérations ou sur la valeur du résidu. Parfois, l'augmentation de K est aussi considérée comme une condition d'arrêt car cela indique que l'algorithme ne converge pas vers un minimum.

1.4 Exemple d'implémentation

Une implémentation en Python et CUDA a été réalisée au sein du module open-source Crappy. Ce module Python permet le pilotage d'essais complexes multi-instrumentés [2]. Il est disponible sous licence GPLv2 [1].

Le calcul sur CPU est inefficace pour traiter de grands volumes de données. À l'inverse, le GPU est conçu pour effectuer du traitement d'images et gérer les données en parallèle. Afin d'atteindre des performances proches du temps réel, il a été choisi d'implémenter cette méthode de corrélation avec une accélération par carte graphique à l'aide du langage CUDA. Le module Python PyCUDA permet d'appeler des routines pour GPU à partir du langage Python. C'est ce qui a été utilisé pour implémenter cette méthode au sein de *Crappy*, le module Python de pilotage d'essais utilisé au LaMcube.

Dans cette méthode, les deux étapes nécessitant le plus d'opérations sont l'écriture de l'image $D(x) = F(x) - G(x + u(x))$ et le calcul du vecteur S . Afin de parcourir les images aussi rapidement que possible dans l'algorithme, il a été choisi de tirer parti des unités de texture. Celles-ci présentent entre autres l'intérêt de fournir une accélération matérielle pour l'interpolation bilinéaire. La lecture d'un point à coordonnées non entières ne nécessite plus d'effectuer quatre accès mémoire et devient aussi rapide qu'un seul accès. Si l'interpolation bilinéaire peut paraître simpliste, il existe des méthodes permettant d'effectuer une interpolation dite "quasi-bicubique" en seulement quatre interpolations bilinéaires.

Afin de calculer S , il faut parcourir l'ensemble des tables $\nabla F \cdot U_i$ multipliées par D et en effectuer la somme. Il s'agit d'opérations de réduction. Afin d'accélérer ces opérations de réduction, il faut éviter les accès mémoire inutiles et paralléliser les opérations tant que possible. Par ailleurs, il est important de tirer parti de la mémoire partagée du GPU, présentant une latence nettement inférieure à la mémoire RAM de la carte graphique. Le module PyCUDA propose des kernels de réduction optimisés et c'est la solution qui a été retenue pour calculer les composantes du vecteur direction de recherche.

Notre implémentation accélérée par GPU permet d'effectuer le calcul de corrélation en temps réel jusqu'à 30Hz sur une machine grand public avec des images 8bits de 4 Megapixels.

La première application de cet outil est la surveillance du bon déroulement d'un essai. En effet, avec un seuil de détection inférieur au dixième de pixel, un grossissement de 1x et un pixel de $5\mu m$, il est possible de détecter de façon fiable et instantanée un déplacement de $0.5\mu m$ ainsi que sa direction. En mesurant le mouvement de corps rigide, cela permet de très rapidement identifier les mouvements parasites liés à la machine ou au montage dès leur apparition comme les jeux mécaniques ou le glissement de l'éprouvette et cela bien avant que cela ne soit observable à l'œil nu. Cela permet d'interrompre l'essai

si besoin, potentiellement avant d'endommager le matériau. Les données suivantes présentent un essai de traction sur un matériau polycarbonate avec une machine bi-axiale plane à vitesse constante. La machine et le montage utilisés sont présentés sur la figure 1.

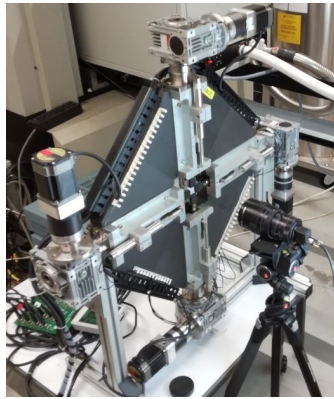


FIGURE 1 – Montage utilisé pour les essais bi-axiaux

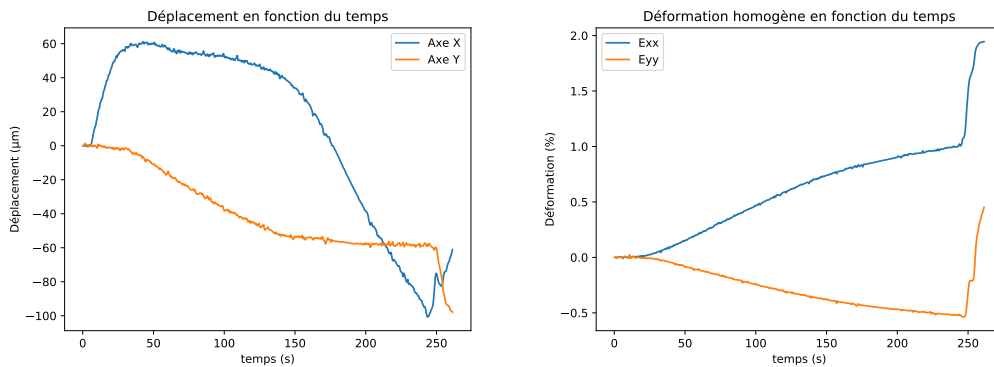


FIGURE 2 – Essai de traction sur un matériau polycarbonate suivi par corrélation en temps réel

La figure 2 permet d'observer un jeu machine important au début de l'essai sur l'axe X qui induit un fort déplacement selon cet axe et empêche l'augmentation de E_{xx} tant que le jeu n'est pas rattrapé. En plus de créer un délai avant l'apparition de la déformation, cela implique un décalage de l'éprouvette qui peut causer une dissymétrie non souhaitable au cours de l'essai bi-axial. Afin de pallier ce problème, le pilotage a été modifié pour prendre en compte ce déplacement et le corriger en imposant une commande de vitesse différentielle entre les moteurs opposés. Ainsi, la position de la zone d'observation est asservie pour rester centrée au cours de l'essai, évitant les déplacements pouvant introduire un cisaillement non désiré.

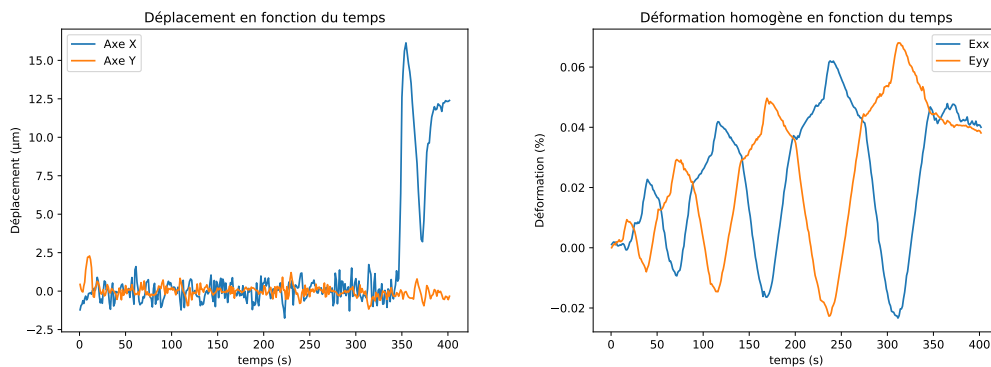


FIGURE 3 – Essai de traction sur carbone-époxy piloté par corrélation

Dans les données représentées sur la figure 3, le déplacement de l'éprouvette observé au début de l'essai est rapidement corrigé et est de l'ordre de quelques micromètres, jusqu'à l'apparition à 350s d'un glissement important ne permettant pas d'atteindre l'effort cible, entraînant la fin de l'essai.

Un second exemple d'application est la localisation et la mesure du maximum de déformation au cours de l'essai. Cela permet notamment de réaliser des essais cycliques en maîtrisant la déformation appliquée sur l'échantillon. Cette méthode de corrélation en temps réel est aussi parfaitement adaptée dans le cas d'une décomposition du champs de déplacement par POD. En effet, il est possible d'extraire d'essais ou de simulations une base de champs de déplacement pertinente pour réaliser la projection. Par ailleurs, le résidu donne une évaluation de l'écart à cette projection, permettant de vérifier systématiquement la pertinence des hypothèses choisies pour la projection.

2 Application au pilotage d'un essai mécanique

La corrélation d'images en temps réel est un outil qui fournit des informations qui ne sont pas nécessairement accessibles avec d'autres moyens de mesure. Cette section présente ce que cette technique peut apporter pour le pilotage d'essais mécaniques.

2.1 Choix de la base de projection

La méthode présentée ici nécessite de connaître a priori la forme du champ cinématique. En effet, le résultat de la corrélation va très fortement dépendre de la base de projection choisie, eq (3). Quelle que soit la base choisie, sa pertinence est simple à quantifier en analysant le résidu. Il s'agit d'un excellent indicateur pour comparer différentes bases et permettant de déterminer celles qui sont les plus adaptées.

Pour déterminer l'espace de projection, il existe plusieurs approches : la plus simple se résume au choix d'une base intuitive. Par exemple les 3 champs suivants permettent de prendre en compte les mouvements de corps rigide 2D (déplacements en x et y , rotation).

$$U_x \begin{pmatrix} i \\ j \end{pmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; U_y \begin{pmatrix} i \\ j \end{pmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; U_r \begin{pmatrix} i \\ j \end{pmatrix} = \begin{bmatrix} j \\ -i \end{bmatrix} \quad (8)$$

Pour simplifier l'expression des champs, ceux-ci sont exprimés dans un système de coordonnées normalisé sur la taille de l'image et prenant son origine au centre de l'image. U_x et U_y représentent respectivement les déplacements selon les directions x et y . U_r est la linéarisation de la rotation dans le sens trigonométrique. Il est possible d'ajouter les déformations uniformes selon x et y , U_{exx} et U_{eyy} et une expression du cisaillement symétrique U_{exy} dont les expressions sont proposées ci-dessous :

$$U_{exx} \begin{pmatrix} i \\ j \end{pmatrix} = \begin{bmatrix} i \\ 0 \end{bmatrix}; U_{eyy} \begin{pmatrix} i \\ j \end{pmatrix} = \begin{bmatrix} 0 \\ j \end{bmatrix}; U_{exy} \begin{pmatrix} i \\ j \end{pmatrix} = \begin{bmatrix} j \\ i \end{bmatrix} \quad (9)$$

Ces champs peuvent permettre de piloter l'essai par des grandeurs scalaires, comme dans le cas d'un essai piloté par une jauge de déformation. La différence est la possibilité d'obtenir une mesure globale en adaptant le champ de la caméra à la zone d'étude. Ce type de pilotage a déjà été réalisé pour établir un asservissement de E_{xx} et E_{yy} lors d'essais bi-axiaux [10]. Il existe de multiples variantes possibles comme par exemple la prise en compte des rotations hors plan avec des termes en produits ij . Par exemple, la CIN a été utilisée pour piloter les six degrés de liberté d'un hexapode dans l'espace [6]. Lors de la construction de la base, il faut s'assurer de ne pas inclure de termes redondants. Par exemple la rotation linéarisée et les cisaillements E_{xy} et E_{yx} donnent un degré de liberté supplémentaire empêchant la convergence de l'algorithme. Une grande partie des phénomènes observés lors d'essais mécaniques pourront être modélisés par une base de champs analytiques.

Ce principe peut être étendu en proposant l'utilisation d'une base polynomiale. Cette approche simpliste a montré des résultats globalement satisfaisants. En effet, sur une zone petite à l'échelle de la structure, la prise en compte des déformations d'ordre 1 ou 2 est souvent suffisante et présente des résidus comparables à la corrélation locale.

2.2 Application de la POD

Pour générer une base adaptée à un essai particulier, une autre approche consiste en l'utilisation de la *Proper Orthogonal Decomposition* (POD) [7]. Cette méthode a pour objectif de chercher l'espace vectoriel limité à l dimensions pour représenter de façon optimale des vecteurs d'un espace à plus de l dimensions.

La POD est utilisée en mécanique des fluides sur des champs obtenus par PIV [4] mais son utilisation en mécanique du solide est novatrice. Cependant, la PGD a déjà été utilisée dans des algorithmes de corrélation [9], celle-ci vise à décomposer chaque champs en une somme de fonction à variables séparables.

Le principe de la POD peut être exprimé de la façon suivante : Soit x un vecteur aléatoire exprimé dans une base orthonormée. L'objectif est de trouver la base limitée à l vecteurs qui minimise la différence entre x et sa projection dans la nouvelle base. Cela revient à chercher les vecteurs Φ_i qui minimise l'espérance de la différence entre x et sa projection, tel que décrit dans l'équation (10).

$$\underset{\Phi_i}{\operatorname{argmin}} (\varepsilon^2(l)) = E(\|x - x(\Phi(l))\|^2) \quad (10)$$

$x(\Phi(l))$ est la projection de x dans la base Φ limitée aux l premiers vecteurs. Il a été montré que cette base peut être construite en diagonalisant la matrice de covariance du vecteur x , triant les colonnes par valeurs propres décroissantes et limiter aux l premières composantes. En connaissant un grand nombre d'échantillons du vecteur, il est alors possible d'estimer la matrice de covariance et donc de construire la base optimale.

2.3 Génération de la base par simulation numérique

Afin de générer une base de champs cinématiques pour le suivi d'un essai bi-axial sur un matériau composite, un ensemble de simulations numériques représentant l'essai ont été réalisées. La simulation a été effectuée sur un maillage relativement grossier, l'objectif étant de faire varier de façon aléatoire un ensemble de paramètres de l'essai et de multiplier les simulations.

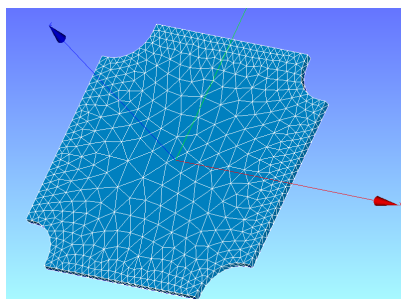


FIGURE 4 – Maillage utilisé pour les simulations

Les paramètres variables sont les conditions aux limites en déplacement correspondant au déplacement appliqué par la machine d'essai, mais aussi des déplacements parasites pouvant provenir des imperfections du montage. Le modèle de la simulation est un modèle élastique linéaire orthotrope et les différents modules du matériau sont eux aussi des paramètres variables de l'essai. Les simulations numériques ont été effectuées avec le logiciel libre Code_Aster. Un script Python a été réalisé afin d'automatiser et de paralléliser les simulations. Cela permet d'obtenir rapidement un ensemble conséquent de champs cinématiques. En effet, sur un PC grand public, 10 minutes suffisent pour réaliser plus de 400 simulations.

2.4 La POD sur des champs cinématiques

Les déplacements aux nœuds sont représentés sous la forme d'un vecteur et les multiples simulations effectuées servent d'échantillons aléatoires pour l'application de la POD.

Avant de travailler sur les simulations décrites précédemment, des tests ont été effectués avec des simulations simplifiées. Le premier cas de test comporte seulement 4 paramètres aléatoires qui correspondent aux déplacements dans les mors de la machine bi-axiale et le modèle utilisé est élastique isotrope. Dans ce cas, la POD limitée à 4 vecteurs représente parfaitement l'ensemble des champs cinématiques obtenus avec la simulation. Le cinquième vecteur issu de la POD possède une valeur propre 10^{12} fois plus faible que la précédente, il est donc possible de représenter l'ensemble des simulations avec seulement 4 composantes sans erreur notable. En ajoutant 4 déplacements supplémentaires, il est à nouveau possible de représenter l'ensemble des simulations avec 8 champs. Dans les deux cas précédents, il existe même une bijection entre les champs de déplacement et les paramètres de l'essai. Ce n'est plus le cas lorsque les paramètres matériaux sont aussi des grandeurs aléatoires de la simulation.

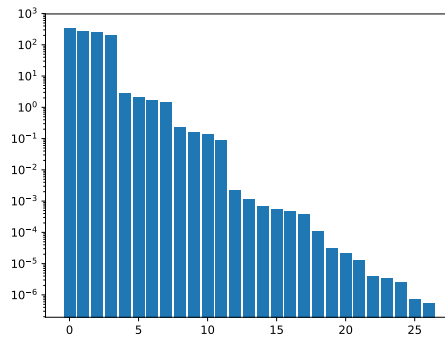


FIGURE 5 – Valeurs propres associées aux premiers vecteurs propres issus de la POD sur les simulations avec matériau variable

Dans le cas de la simulation décrite dans le paragraphe 2.3, les 27 premières valeurs propres sont représentées sur la figure 5. Quelques champs sont représentés sur la figure 6. Il n'existe pas de distinction majeure permettant de choisir le nombre de vecteurs propres significatifs et il devient plus délicat de placer la limite du nombre de vecteurs à conserver.

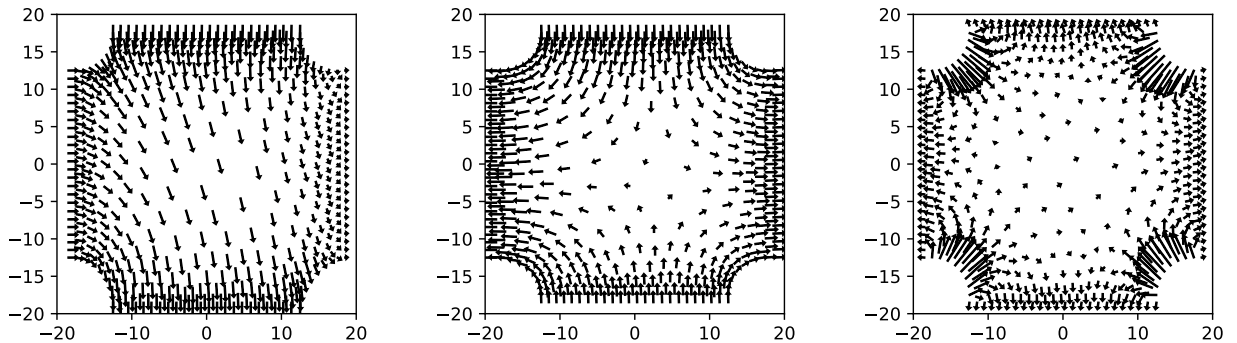


FIGURE 6 – Premier, second et neuvième vecteurs propres obtenus par POD

2.5 Application pour l'identification de propriétés ou de grandeurs internes

Une des utilisations envisagée de cet outil est la quantification de l'écart à un comportement matériau défini. Par exemple, si une base permet de représenter avec une précision acceptable l'ensemble des champs cinématiques obtenus par déformation élastique linéaire, l'augmentation du résidu peut illustrer une sortie du cadre choisi et donc l'apparition d'un comportement non pris en compte par le modèle. Aussi, l'utilisation de champs cinématiques réels comme point de départ pour la POD pourra certainement permettre l'identification de plages de comportement matériau voire même aller jusqu'à la quantification de variables internes.

3 Conclusion et perspectives

Ce travail exploratoire des techniques de corrélation en temps réel ouvre des perspectives riches en matière de pilotage d’essais mécaniques. Couplée à la POD, cette technique pourra permettre de créer un espace vectoriel dans lequel les grandeurs à observer dans un essai sont directement accessibles, comme par exemple l’endommagement d’un matériau. Il sera alors possible de mettre en place une boucle de pilotage directement sur les grandeurs étudiées. Cependant, la méthode proposée ici présente un certain nombre de limites. En particulier l’impossibilité de la prise en compte de discontinuités sans savoir au préalable leur emplacement. Une possibilité pour pallier ce problème serait l’ajout dynamique de nouvelles composantes à la base de projection en effectuant la POD pendant l’essai.

Références

- [1] V. Couty, C. Martel. *Command and Real-time Acquisition in Parallelized Python*, <https://github.com/LaboratoireMecaniqueLille/crappy>.
- [2] V. Couty. *CRAPPY : un module pour le pilotage de systèmes multi-instrumentés*, PyconFR 2018, IMT Lille-Douai, 2018.
- [3] H. Dianyuan. *Comparison of commonly used image interpolation methods*, Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering, Atlantis Press, 1556-1559, 2013.
- [4] L. Graftieaux, M. Michard, N. Grosjean. *Combining PIV, POD and vortex identification algorithms for the study of unsteady turbulent swirling flows*, Measurement Science and Technology vol. 12, 1422–1429, 2001.
- [5] F. Hild, S. Roux. *Measuring stress intensity factors with a camera : Integrated digital image correlation (I-DIC)*, Comptes Rendus Mécanique, 8-12, 2006.
- [6] J. Le Flohic, V. Parpoil, S. Bouissou, M. Poncelet, H. Leclerc. *A 3D Displacement Control by Digital Image Correlation for the Multiaxial Testing of Materials with a Stewart Platform*, Experimental Mechanics 54, 817–828, 2014.
- [7] Y.C. Liang, H.P. Lee, S.P. Lim. *Proper Orthogonal Decomposition and its applications—part I : Theory*, Journal of Sound and Vibration, 527-544, 2002.
- [8] M. Pharr, R. Fernando. *GPU Gems 2 : Programming Techniques for High-Performance Graphics and General-Purpose Computation*, 2005.
- [9] J.-C. Passieux, J.-N. Périé. *High resolution digital image correlation using proper generalized decomposition : PGD-DIC*, International Journal for Numerical Methods in Engineering vol. 92, 531-550, 2012.
- [10] M. Poncelet, G. Barbier, B. Raka, S. Courtin, R. Desmorat, J.C. Le-Roux, L. Vincent. *DIC-aided biaxial fatigue tests of a 304L steel*, EPJ Web of Conferences 6, 2010.
- [11] D.F. Shanno. *Conditioning of quasi-Newton methods for function minimization*, Mathematics of computation 111, American Mathematical Society, 647-656, 1970.