



HAL
open science

Escale : une plateforme ouverte pour le dialogue essai-modèle-calcul

Sylvia Feld-Payet, Thomas Fourest, Pascal Bouda, Jean-Didier Garaud,
Francesco Bettonte, Aly Tireira, Yves Le Sant, Guy Le Besnerais, Steve Belon

► **To cite this version:**

Sylvia Feld-Payet, Thomas Fourest, Pascal Bouda, Jean-Didier Garaud, Francesco Bettonte, et al..
Escale : une plateforme ouverte pour le dialogue essai-modèle-calcul. CSMA 2022 15ème Colloque
National en Calcul des Structures, May 2022, Presqu'île de Giens, France. hal-03687482

HAL Id: hal-03687482

<https://hal.science/hal-03687482>

Submitted on 3 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Escale: une plateforme ouverte pour le dialogue essai-modèle-calcul

S. Feld-Payet¹, T. Fourest², P. Bouda^{2*}, J.D. Garaud¹, F. Bettonte^{1*}, A. Treira^{1*},
Y. Le Sant^{3*}, G. Le Besnerais⁴, S. Belon^{2*}

¹ DMAS, ONERA, Université Paris Saclay, F-92322 Châtillon - France {sylvia.feld-payet}@onera.fr

² DMAS, ONERA, F-59014 Lille - France

³ DAAA, ONERA, Université Paris Saclay, F-92190 Meudon - France

⁴ DTIS, ONERA, Université Paris Saclay, F-91123 Palaiseau, France

* avant départ de l'ONERA

Résumé — La plateforme logicielle Escale a été développée pour répondre au besoin de capitalisation et mutualisation des outils dans le domaine du dialogue essai-modèle-calcul pour la mécanique du solide. Elle comprend des modules de traitement et analyse de données permettant la génération d'images de synthèses, la mesure de champs cinématiques, la détection de fissures, l'identification de paramètres de loi de comportement et la confrontation essais-calculs. En outre, d'autres modules permettent le dialogue entre ces modules internes et des logiciels externes.

Mots clés — dialogue essai-modèle-calcul, mesure de champs denses, Python.

1 Introduction

Pour s'assurer qu'une structure industrielle réponde parfaitement aux besoins exprimés, il faut s'intéresser à toutes les étapes de sa vie : depuis le procédé de fabrication à la ruine complète de la structure en passant par sa tenue en service. Tout ceci nécessite de pouvoir connaître les matériaux utilisés et comprendre les phénomènes qui pilotent leur comportement afin de pouvoir modéliser avec le degré de précision voulu la réponse de ces structures.

La première étape consiste ainsi à étudier le comportement mécanique du matériau sur des éprouvettes de petites dimensions. Pour analyser ces essais, en particulier en présence de phénomènes localisés, l'utilisation de méthodes optiques pour la mesure de champs cinématiques s'est imposée depuis le début des années 1980, notamment grâce à la Corrélation d'Images Numériques (CIN) [10, 11]. L'enjeu consiste à mesurer des champs de déplacement sur la surface (voire dans le volume) d'éprouvettes sous sollicitations mécaniques à partir d'images acquises à différents stades du chargement. Un atout spécifique de cet outil est qu'il exploite des images numériques, le plus souvent acquises par des moyens optiques qui sont par essence sans contact. Ils permettent donc d'effectuer des mesures à distance alors que l'éprouvette est soumise à des conditions agressives et expérimentalement difficiles telles que les très hautes ou basses températures et ils peuvent, en fonction des capteurs optiques utilisés, être exploités pour toutes les échelles d'espace, de temps et de température.

Dans un deuxième temps, les paramètres identifiés (qui peuvent caractériser le matériau, la structure elle-même, ou les conditions de chargement) sont généralement utilisés dans une simulation éléments finis dont les résultats sont comparés à ceux de l'essai sur éprouvette. Une bonne correspondance essai-calcul permet d'une part de valider le modèle et la méthodologie de calcul et d'autre part d'envisager d'utiliser ces outils de calcul pour des simulations permettant de prédire le comportement de structures entières. Le caractère prédictif de ces modèles est important ; il permet de diminuer le nombre d'essais sur structure (moyennant l'augmentation du nombre d'essais sur éprouvettes structurelles).

Cependant, la confrontation des calculs aux mesures denses issues de ces techniques de corrélation d'image nécessite un certain nombre d'outils. Le développement sans concertation ni mutualisation de ces outils en parallèle dans des entités différentes est contre-productif et constitue un frein aux échanges sur des problématiques communes. Au contraire, l'approche envisagée avec la plateforme Escale est de capitaliser et mutualiser les développements afin de concentrer les ressources sur le développement d'outils nouveaux. Pour ce faire, l'idée est, d'une part, de mettre à disposition les outils déjà développés, et d'autre part, de faciliter les échanges avec les outils externes existants.

2 La plateforme Escale

2.1 Objectif

La plateforme logicielle Escale a été initialement développée à l’Onera pour ses besoins propres dans le domaine du dialogue essai-modèle-calcul pour la mécanique du solide [5]. Elle a été conçue afin de favoriser le développement et la capitalisation des outils développés pour la mesure de champs par analyse d’images. Elle s’ouvre en 2022 au milieu académique français qui partage, dans ce domaine, les mêmes besoins de capitalisation et mutualisation des outils.

2.2 Description générale

La plateforme de dialogue essais/ modélisation/ simulation Escale consiste en une couche de couplage entre différents composants logiciels. Elle comprend :

- des modules de traitement et analyse de données,
- des modules permettant le dialogue entre ces modules internes et des logiciels externes.

Les modules de traitement et analyse de données, décrits dans la partie 3, permettent notamment :

- de générer des images synthétiques déformées par un champ de déplacement éléments finis ;
- de calculer des champs de déplacement cinématiques denses à partir d’images ;
- de détecter des fissures ;
- d’identifier des paramètres de loi de comportement ;
- de comparer quantitativement les résultats d’essais à ceux des calculs.

Ces modules sont à combiner suivant les besoins de l’utilisateur, comme illustré figure 1, afin de mener une analyse avancée des essais en tirant parti de la modélisation et/ou de la simulation.

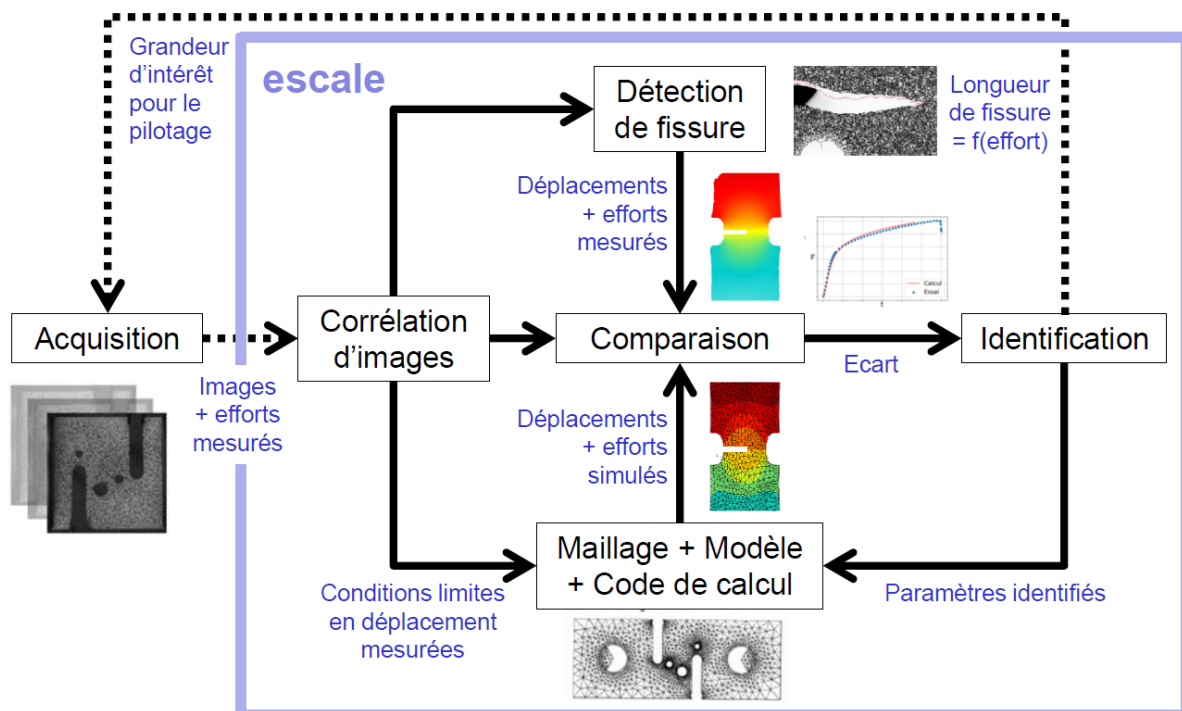


FIGURE 1 – Exemple de combinaison de composants dans Escale.

Il est par exemple possible de faire d’abord appel à un composant de corrélation d’images qui va fournir un champ de déplacement dense. Puis ce champ de déplacement pourra être comparé au champ de déplacement simulé, en utilisant au besoin une procédure de recalage maillage-image. La différence des deux champs peut enfin servir lors d’une procédure d’optimisation d’une fonction coût afin d’identifier les paramètres du modèle de comportement. Afin de rendre la comparaison plus représentative, il est également possible d’imposer en entrée du calcul les conditions aux limites en déplacements mesurées. Enfin, il est non seulement envisageable de comparer des champs denses, mais aussi des mesures plus

ponctuelles ou même scalaires, comme par exemple des positions de front de fissure, ou des longueurs de fissures.

2.3 Comment s'en servir ?

La combinaison des composants se fait grâce au script utilisateur, dont la structure générale est décrite par la figure 2.

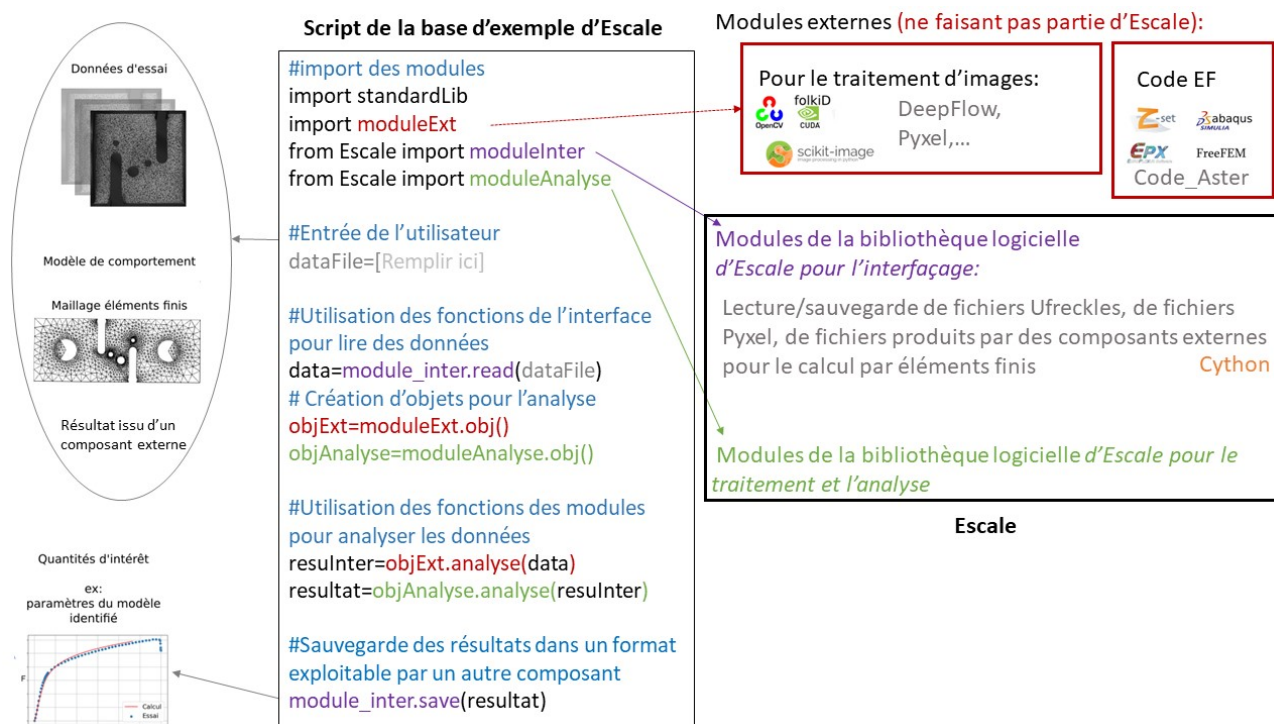


FIGURE 2 – Structure générale d'un script utilisateur.

Les composants externes peuvent facilement être utilisés en combinaison avec les composants internes d'Escale, soit via une interface qui va faire appel à une fonction du composant externe (comme pour les logiciels Folki_D [4, 8, 9] ou Zset [19]) notamment à l'aide du langage Cython, soit en utilisant directement les fonctions des composants externes s'ils sont également en Python (comme pour la librairie Pyxel [17]), ou encore en important les résultats de l'analyse faite par le composant externe au format Numpy ou HDF5 (comme pour Ufreckles [18]).

Afin de faciliter l'établissement de ces scripts, l'utilisateur peut se référer à la base d'exemples documentée. Cette base permet non seulement de mettre en œuvre les fonctions d'Escale, mais également de montrer comment faire fonctionner Escale avec les outils partenaires tels que Folki_D, Ufreckles et Pyxel.

2.4 Qualité

La base d'exemples sert aussi de base de tests afin de vérifier le comportement attendu de la part du logiciel et d'assurer la non-régression du code.

Afin d'avoir un code uniforme respectant une convention de codage (PEP8 [16]), qui sera plus facile à maintenir et plus agréable à lire, Escale s'appuie sur l'utilisation de linters. Ces outils permettent d'associer une note à la qualité du code et toute nouvelle contribution au code se doit de ne pas diminuer la note courante pour être acceptée.

Pour le gestionnaire de code source, qui définit la façon dont les co-développeurs s'organisent pour contribuer au projet, c'est le logiciel de versionnage Git qui a été retenu. Les systèmes associés permettent aux développeurs travaillant en équipe d'éditer en même temps le code source, apporter des modifications et garder une vue d'ensemble de toutes les modifications réalisées. Chaque modification peut être suivie avec exactitude et éventuellement annulée. En particulier, le système GitLab a été préféré à GitHub,

notamment parce qu'il autorise un nombre illimité de référentiels gratuits, de « branches protégées » (dont l'accès est réservé à certains utilisateurs) associées et de développeurs par référentiel privé.

Enfin, la documentation étant un outil indispensable pour une maintenance facile du code et une communication efficace, le logiciel Escalé s'accompagne d'une documentation wiki des exemples et d'une documentation du code décrite en en-tête de chaque fonction et compilée par l'outil Sphinx.

3 Les différents modules de traitement et d'analyse d'Escalé

3.1 Simulateur d'images synthétiques déformées

De manière générale, les caméras utilisées en mesure de champs cinématiques possèdent des propriétés (fréquence, résolution, bruit, ...) qui influencent les résultats obtenus. Il s'avère aujourd'hui indispensable de prendre en compte un maximum de ces propriétés pour reproduire le plus fidèlement possible la prise des clichés de la scène d'essai, lors de la préparation d'essais complexes. Pour ce faire, un simulateur d'images synthétiques déformées [3] permet de reproduire l'étape de digitalisation par les caméras filmant la scène. Ce simulateur génère, à partir des résultats d'un calcul éléments finis et d'une texture (analytique ou d'une image) initiale, une séquence d'images reproduisant le suivi de la déformation de l'éprouvette sous chargement.

Un exemple d'application relatif à l'impact, à 100 m/s, d'un cylindre sur la tranche d'une éprouvette plate de longueur 32 mm est ici présenté. La texture initiale est une grille de pas de 0.15 mm (voir figure 3, à gauche). Elle est déformée à partir des résultats d'une simulation éléments finis réalisée avec le code de calcul de dynamique transitoire Europlexus [15] (voir figure 4, à droite), en prenant en compte les paramètres de la caméra envisagée pour le suivi d'essai. L'éprouvette et la grille sont montrés dans une configuration déformée figure 3, à droite.

Disposant des images déformées simulées, il est possible d'effectuer les traitements envisagés (mesure, identification, ...) afin de valider la faisabilité de l'essai.

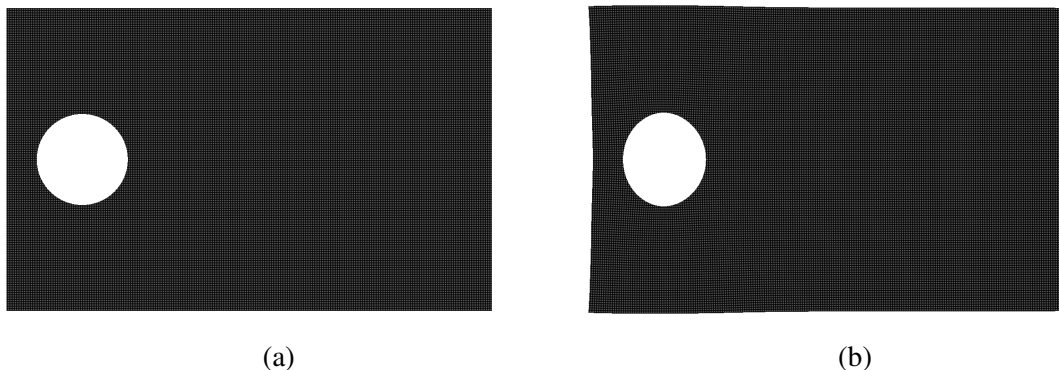


FIGURE 3 – Images de la grille de 0.15 mm déposée sur une éprouvette perforée de longueur 32 mm. (a) Configuration initiale. (b) Configuration déformée 15 micro-secondes après l'impact.

3.2 Composants de mesure de champs cinématiques denses

Trois composants pour la mesure de champs cinématiques denses en mono-vision sont disponibles dans Escalé :

- un composant utilisant la méthode des grilles [1]. Cette dernière a été par exemple appliquée sur les images générées par le simulateur d'images déformées (décrit en 3.1) pour obtenir le champ de déformation en figure 4, à gauche.
- un composant de corrélation locale s'appuyant sur l'algorithme Folki développé par l'Onera, dénommé *Folki-P mono-vision*. Il s'agit d'une version Python pour la mono-vision de l'algorithme `Folki_D`¹.

1. `Folki_D` utilise les capacités du langage cuda pour atteindre un niveau de performance exceptionnel en terme de ratio quantité de données traitées/temps de calcul et permet en outre de faire de la stéréo-corrélation. Cette version cuda est accessible via Escalé, mais doit faire l'objet d'une demande de licence séparée.

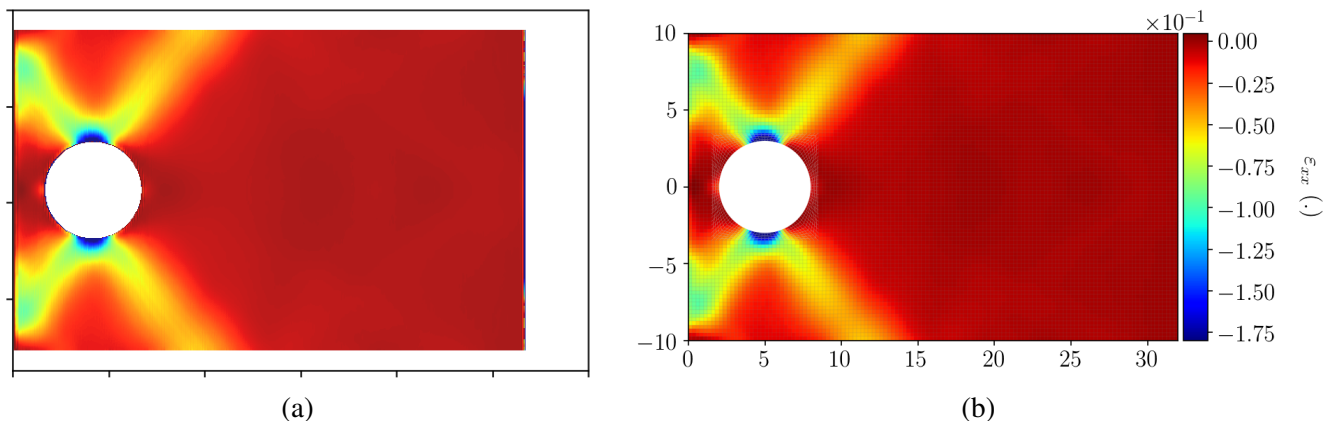


FIGURE 4 – Champ de déformation pour un impact à 100m/s dans la configuration de référence, 15 micro-secondes après l’impact : obtenus par (a) par le traitement par la méthode grilles des images générées par le simulateur (b) la simulation éléments finis.

- un composant de corrélation globale s’appuyant sur la méthode des éléments finis et en particulier sur le code de calcul Zset. Cet algorithme est décrit dans [5] et est également accessible via une interface graphique pour faciliter son utilisation.

En plus des outils d’Escale, il est également possible de faire appel à des bibliothèques Python externes, tels que la bibliothèque OpenCV. Cette dernière donne notamment accès à l’algorithme de type variation totale DeepFlow développé par l’INRIA [12] qui peut être utilisé pour la détection de fissure (voir partie 3.3). Il est également possible d’utiliser les outils des partenaires co-développeurs, tels que UFreckles ou Pyxel, dont les résultats peuvent être ensuite exploités par un autre composant de la plateforme.

3.3 Détection de fissures

Dans Escale, sont également disponibles des outils pour la détection de fissure. A partir d’un champ scalaire bien choisi (erreur de corrélation ou quantité lié au gradient des déplacements), il est possible de construire une approximation du chemin de fissure par une méthode de moyenne pondérée : voir figure 5, à gauche. Ce chemin potentiel peut ensuite être exploité afin de rechercher la position du front de fissure. Le critère de détection empirique de la position actuelle du front de fissure proposé dans [6] est disponible. Associé à la carte de maximum de gradient fournie par DeepFlow, il permet de traiter aussi bien la propagation de fissure fragile que ductile, pour le mode I ou le mode II.

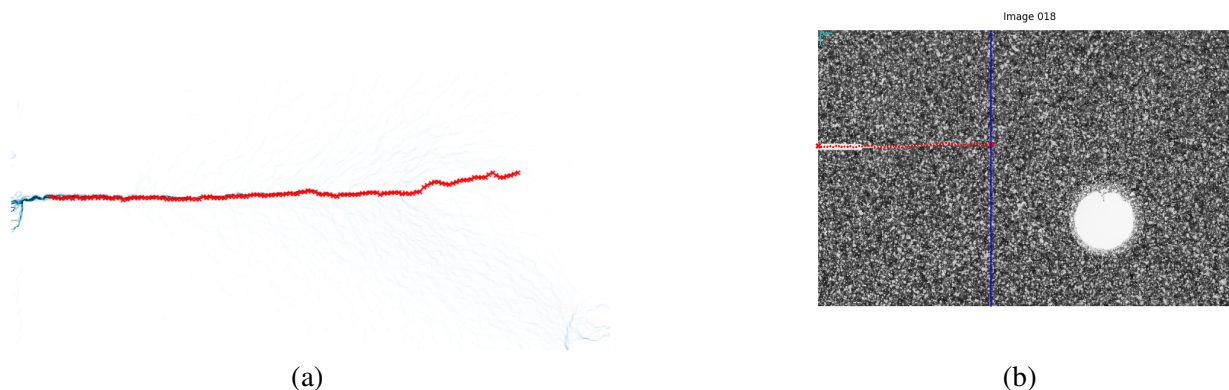


FIGURE 5 – Détection de fissure sur une plaque trouée en acier utilisée pour le GDR Fatacrack.(a) exploitation de la carte de maximum de gradient fournie par DeepFlow pour trouver un chemin potentiel grâce à une moyenne pondérée toutes les 8 colonnes (zoom).(b) en rouge, fissure déterminée grâce au critère proposé dans [6] projetée sur la configuration courante, en bonne correspondance avec la position horizontale du front de fissure obtenue expérimentalement.

3.4 Identification

Deux méthodes d'identification de paramètres matériau, notamment décrites dans l'ouvrage de référence [7], sont disponibles dans Escale :

- La méthode de recalage du modèle éléments finis (ou *FEMU* pour *Finite Element Methode Updating* en anglais), qui consiste à minimiser une fonction coût établie par différence entre les résultats d'essai et les résultats de calculs éléments finis correspondants aux valeurs successives des paramètres recherchés. L'implantation de cette méthode dans Escale a notamment été mise en œuvre lors de la thèse de Francesco Bettonte [2] pour l'identification des paramètres d'un modèle de comportement élasto-visco-plastique endommageable à partir d'essais sur différentes éprouvettes en Inconel 625.
- La Méthode des Champs Virtuels (MCV), une méthode sans calcul élément finis qui est basée sur le principe des travaux virtuels. Pour les lois de comportement non-linéaires elle consiste de manière générale en la minimisation d'une fonction coût établie par différence entre le travail virtuel interne (calculé à partir du champ de contrainte lui-même calculé à partir des déformations mesurées pour un set de paramètres) et le travail virtuel externe (calculé à partir de l'effort expérimental). Ces travaux sont dits virtuels car ils font intervenir des champs virtuels de déplacement et de déformation qui peuvent être vus comme des fonctions de pondération. L'implémentation de cette méthode dans Escale a notamment été mise en œuvre lors de la thèse de Pascal Bouda [13] pour l'identification de paramètres de comportement visco-plastique sur des essais d'impact.

3.5 Outils de pré/post-traitement pour le dialogue essai-calcul

Escale contient également un certain nombre d'outils pour aider à comparer des résultats d'essais stockés à l'échelle du pixel (dans le cas où une méthode de corrélation d'images locale serait préférée à une méthode globale par éléments finis), à des résultats de calcul stockés au niveau d'éléments finis. Il est par exemple nécessaire de pouvoir aligner le maillage à l'image de l'éprouvette pour pouvoir comparer les déplacements au même point et dans la même unité. Pour cela, Escale contient deux méthodes d'alignement maillage-image s'appuyant respectivement sur une homographie ou une solution de problème de perspective-n-point. A partir de là, il est par exemple possible de placer une jauge virtuelle sur une carte de déplacement afin de comparer le résultat au calcul.

4 Conclusion et perspectives

Escale est une plateforme logicielle dédiée au dialogue essai-modèle calcul en mécanique du solide. Elle utilise Python pour l'assemblage de composants, ce qui lui confère l'avantage d'être très modulaire. Les utilisateurs visés sont des utilisateurs avertis appartenant au milieu de la recherche. A partir de 2022, ils pourront profiter des fonctions d'Escale et/ou y contribuer grâce à une diffusion sous licence libre. L'ajout de contributions et la politique de développement seront gérés par un consortium composé de l'Onera, du LaMCube/Centrale Lille, du GeM/Centrale Nantes, de l'ICA/INSA Toulouse, du CEA Saclay et des Arts et Métiers d'Aix-en-Provence. L'objectif de cette ouverture est que la communauté académique française intéressée par la mesure de champs par analyse d'images en mécanique du solide puisse disposer d'un outil commun qui permette de faciliter les échanges et capitaliser les développements.

A l'avenir, afin d'aider la poursuite de l'interfaçage d'Escale avec des outils logiciels externes, il est envisagé de mener une réflexion à l'échelle du consortium sur l'utilisation d'API pour faciliter l'utilisation de composants ayant la même fonction, ainsi qu'une réflexion sur l'utilisation d'un modèle de données adapté pour échanger plus facilement les données entre composants.

Remerciements

Les auteurs souhaitent remercier les membres partenaires du consortium pour leur participation au développement d'Escale : Jean-François Witz au LaMCube/Centrale Lille, Julien Réthoré, Michel Coret et Rian Seghir au GeM/Centrale Nantes, Jean-Charles Passieux, Jean-Noël Périé et Robin Bouclier à

l'ICA/INSA Toulouse, Pascal Bouda au CEA Saclay et Jean-Patrick Goulmy au Arts et Métiers d'Aix-en-Provence, ainsi que François Nicolas et Aurélien Plyer à l'ONERA pour leur assistance avec Folki_D. Les auteurs souhaitent également remercier la société SimulEase pour son aide à la mise en place de la version d'Escale sous GitLab et à l'interfaçage avec Code_Aster.

Références

- [1] C. Badulescu, M. Grédiac, J.-D. Mathias, D. Roux. *A procedure for accurate one-dimensional strain measurement using the grid method*, Experimental Mechanics, DOI 10.1007/s11340-008-9203-8, 2008.
- [2] F. Bettonte. *Développement d'une stratégie d'identification des paramètres par recalage de modèle éléments finis à partir de mesures par corrélation d'images : vers l'application à un modèle d'endommagement non local*, Thèse de doctorat dirigée par J. Besson et S. Feld-Payet, Sciences et génie des matériaux Paris Sciences et Lettres (ComUE), 2017.
- [3] P. Bouda, B. Langrand, D. Notta-Cuvier, E. Markiewicz, F. Pierron. *A computational approach to design new tests for viscoplasticity characterization at high strain-rates*, Computational Mechanics, 64(6) :1639-1654, 2019.
- [4] F. Champagnat, A. Plyer, G. Le Besnerais, B. Leclaire, S. Davoust, Y. Le Sant. *Fast and accurate PIV computation using highly parallel iterative correlation maximization*, Experiments in fluids, 50(4) :1169-1182, 2011.
- [5] J.D. Garaud, S. Feld-Payet, F. Bettonte, A. Tireira, Y. Le Sant, G. Le Besnerais, S. Belon. *escale : plateforme logicielle pour le dialogue essai-calcul*, Proceedings 13ème du Colloque National en Calcul des Structures CSMA, Giens, 2017
- [6] S.Feld-Payet, G. Le Besnerais, V. Bonnard, D. Pacou, L. Thiercelin. *Crack path tracking from full field measurements : A novel empirical methodology*, Strain, 56(2), e12333 10.1111/str.12333, 2020.
- [7] M. Grédiac, F. Hild. *Mesure de champs et identification en mécanique des solides*, Hermes - Lavoisier, 2011.
- [8] G. Le Besnerais, F. Champagnat. *Dense optical flow by iterative local window registration*, IEEE, 137-140, 2005.
- [9] G. Le Besnerais, Y. Le Sant, D. Lévêque. *Fast and dense 2d and 3d displacement field estimation by a highly parallel image correlation algorithm*, Strain, 52 :286-306, 2016.
- [10] B.D. Lucas, T. Kanade. *An iterative image registration technique with an application to stereovision*, Proceedings of the 1981 DARPA Imaging Understanding Workshop, 121-130, 1981.
- [11] M.A. Sutton, W.J. Wolters, W.H. Peters, W.F. Ranson, S.R. McNeill. *Determination of displacements using an improved digital correlation method*, Im. Vis. Comp., 1 :133-139, 1983.
- [12] P. Weinzaepfel, J. Revaud, Z. Harchaoui, C. Schmid. *DeepFlow : Large displacement optical flow with deep matching*, IEEE International Conference on Computer Vision (ICCV), 2013.
- [13] T. Fourest, P. Bouda, L. Fletcher, D. Notta-Cuvier, E. Markiewicz, , F. Pierron, B. Langrand. *Image-based inertial impact test for characterisation of strain rate dependency of Ti6al4v titanium alloy*, Experimental Mechanics, 60(2), 235-248, 2020.
- [14] <https://code-aster.org/spip.php?rubrique1>
- [15] <http://www-epx.cea.fr/>
- [16] <https://www.python.org/dev/peps/pep-0008/>
- [17] <https://github.com/jcpassieux/pyxel>
- [18] <https://zenodo.org/record/1433776>
- [19] <http://www.zset-software.com/>