

## Z-set – suite logicielle pour la simulation des matériaux et structures

JD. Garaud<sup>1</sup>, J. Rannou<sup>1</sup>, C. Bovet<sup>1</sup>, S. Feld-Payet<sup>1</sup>, V. Chiaruttini<sup>1</sup>, B. Marchand<sup>2</sup>, L. Lacourt<sup>2,3</sup>,  
V.A. Yastrebov<sup>2</sup>, N. Osipov<sup>4</sup>, S. Quilici<sup>4</sup>.

<sup>1</sup>ONERA – The French Aerospace Lab, Châtillon France. {jdgaraud,jrannou,cbovet,spayet,vchiarut}@onera.fr

<sup>2</sup>MINES ParisTech, PSL Research University, Centre des Matériaux, CNRS UMR 7633, BP 87, 91003 Evry, France  
{basile.marchand,vladislav.yastrebov,laurent.lacourt}@mines-paristech.fr

<sup>3</sup>CEA-DAM, Centre de Valduc, France,

<sup>4</sup>Transvalor SA, Sophia Antipolis, France, {nikolay.osipov,stephane.quilici}@mines-paristech.fr.

**Résumé** — La suite Z-set est un ensemble d'outils dédiés à la mécanique des matériaux et au calcul de structures. Elle intègre différents modules tels qu'un solveur éléments finis, une bibliothèque d'intégration de loi matériaux, une collection de post-traitements dédiés à l'analyse mécanique, un optimiseur permettant de déterminer les paramètres matériaux d'une loi à partir de données d'essais. Certains de ces modules sont indépendants et peuvent être utilisés par des codes commerciaux sous la forme de plugins. On présentera lors de la session logicielle des éléments développés récemment sur le calcul haute performance, la fissuration et le remaillage adaptatif, la simulation du contact, le couplage de codes et l'interopérabilité.

**Mots clés** — code éléments finis, bibliothèque matériau, durée de vie, calcul parallèle, couplages.

### 1 La suite Z-set

La suite logicielle Z-set est un ensemble d'outils dédiés à la mécanique des matériaux et au calcul de structures sous sollicitations thermo-mécaniques complexes. Elle est conjointement développée par le Centre des Matériaux (Mines ParisTech) et l'Onera. Sa distribution est assurée par Transvalor [1].

Le logiciel poursuit un double objectif :

- capitaliser la recherche réalisée dans ces trois entités, tant dans les domaines des modèles matériaux que du calcul de structures et des méthodes numériques associées ;
- fournir aux utilisateurs – industriels et académiques – un outil de simulation robuste et pérenne.

Développée à l'origine pour les besoins des mécaniciens des matériaux, la composante « lois de comportement » demeure un élément essentiel de la suite logicielle et est disponible à travers son module Z-mat qui fournit une large bibliothèque de comportements matériaux ainsi qu'un langage de haut niveau (ZebFront) pour la définition de nouvelles lois de comportement. Z-mat s'intègre naturellement au solveur éléments finis Zébulon mais est également utilisable sous la forme de plugin dans d'autres codes éléments finis (Abaqus Standard/Explicit, Ansys, Samcef, ...).

Z-mat est complété par le module Z-simopt, un outil d'aide à l'identification de paramètres de lois de comportement à partir de données d'essais.

Au travers du solveur éléments finis Zébulon, de nombreuses fonctionnalités avancées en calcul de structure sont également fournies. Dans ce document, nous présentons principalement deux aspects :

- les outils de remaillage et de propagation de fissures,
- les fonctionnalités de calcul multimodèle : réduction de modèles et couplage de code.

Zébulon comprend également une interface avec Python3 qui permet de réaliser des mises en données très paramétrables et d'interagir avec le code en cours de calcul. Elle offre aussi à l'utilisateur la possibilité de réaliser des prototypes d'algorithmes comme par exemple pour le couplage de codes, l'élaboration de nouveaux schémas d'intégration temporels ou de résolution non-linéaire.

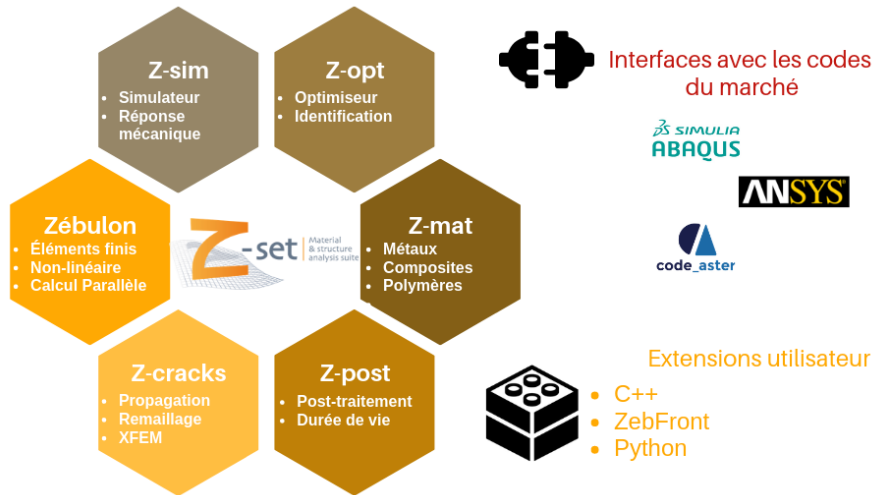


FIGURE 1 – La suite Z-set est composée de différents modules de simulation.

## 2 Z-mat – bibliothèque de lois de comportement

Z-mat est le module de définition des comportements du code Z-set. Il regroupe un module d'intégration de lois de comportement et une bibliothèque de comportements comprenant entre autres l'élasto-visco-plasticité, l'endommagement et les modèles multiéchelles.

La bibliothèque existante comprend plus de 100 000 lignes de code en C++ de modèles matériau. Les modèles sont modulaires ce qui permet une utilisation souple en combinant différentes *briques matériau* (types d'élasticité, critères de plasticité, lois d'écoulement, types d'écrouissage, etc. . .) pour construire un nouveau modèle de façon dynamique. Toutes les combinaisons sont possibles pour décrire le comportement de métaux, polymères, cristaux, inélasticité poreuse, de la mécanique des milieux continus, des sols et des matériaux anisotropes.

De plus, l'utilisateur peut définir son propre comportement, soit en intégrant de *nouvelles briques* dans les classes existantes, soit en décrivant les équations régissant ce comportement à l'aide du langage de haut niveau *ZebFront*, l'intégration étant ensuite réalisée par Z-mat. Z-mat possède également des interfaces avec des logiciels éléments finis du commerce tels que Abaqus, Ansys ou Samcef.

## 3 Z-cracks – une boîte à outils éléments finis pour la discrétisation des discontinuités surfaciques

Les problématiques scientifiques ou industrielles mettant en jeu la présence d'interfaces complexes sont très nombreuses : dans le cadre de la mécanique de la rupture, pour la mise en place de modèles multiéchelles avec représentation de micro-structures, la représentation adaptative de détails géométriques, etc. La discrétisation des domaines étudiés dans de tels contextes est souvent délicate, la présence de ces interfaces, susceptibles d'évoluer dynamiquement, n'étant pas souvent facilement intégrable dans les outils de modélisation génériques disponibles.

C'est pour permettre la discrétisation de tels problèmes qu'a été développé le module Z-cracks. Il permet en particulier de traiter les aspects suivants :

1. **découpe de maillages complexes** pour l'insertion de surfaces et la propagation de fissures avec éventuelles coalescences ;
2. **remaillage adaptatif 3D** basé sur la définition de cartes de taille de maille ou la prescription de zones d'intérêt pour un raffinement local ;
3. **transfert de champs** entre discrétisations, basé sur des algorithmes de type projection  $L^2$  ou collocation pour les champs interpolés, et transferts spécifiques pour les nuages de points (point le plus proche ou avec lissage), le tout associé à des techniques d'équilibrage local et global ;

4. **calcul de forces configurationnelles** pour régir l'évolution géométrique d'interfaces (en fissuration ou en dynamique des dislocations, par exemple);
5. **modularité** des approches permettant leur interfaçage avec plusieurs autres codes éléments finis (Abaqus, ANSYS, Samcef, Code\_Aster ou MEF++).

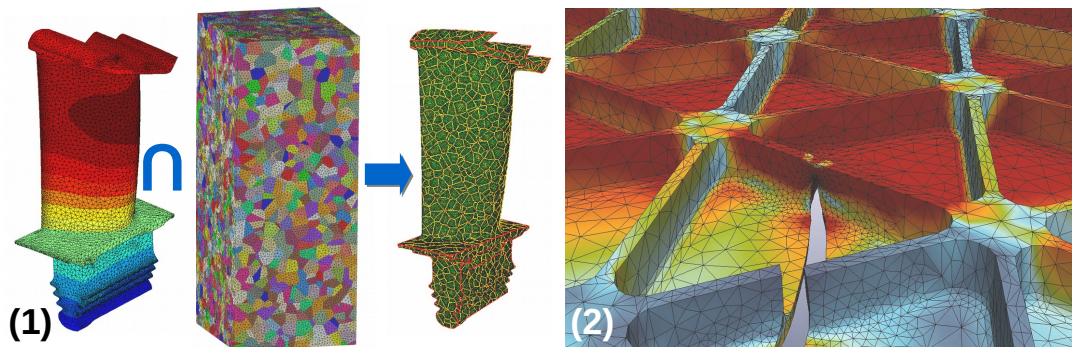


FIGURE 2 – Deux exemples d'application Z-cracks : (1) pour l'insertion d'une micro-structure dans une aube de turbine, (2) pour la propagation de fissure dans une plaque d'aluminium "isogrid".

Ces travaux permettent des applications dans des domaines variés, avec de nombreux partenaires académiques ou industriels (Fig. 2), à travers l'analyse détaillée d'essais de fissuration complexes [2], la prévision de trajectoire de fissures sous des sollicitations multiaxiales complexes de fatigue [3], des approches cohésives adaptatives de la rupture [4], ou des méthodes multiéchelles adaptatives [5].

## 4 Z-post – post-traitements

Z-post est le module de post-traitement de la suite Z-set. Il s'agit d'un outil comprenant une large gamme d'opérations de post-traitement de la simple localisation des points chauds au calcul de durée de vie en chargement cyclique multiaxial en passant par l'évaluation de modèles d'endommagement. Sa souplesse d'utilisation permet de réaliser des traitements complexes sur toutes les grandeurs mécaniques issues d'un calcul éléments finis : quantités nodales, quantités aux points d'intégration, séries temporelles.

La force supplémentaire de Z-post réside dans le fait qu'il s'agit aujourd'hui d'un logiciel indépendant. Il permet de réaliser des post-traitements complexes sur des résultats éléments finis issus d'un calcul Zébulon mais également sur des résultats provenant de codes commerciaux comme Abaqus, Ansys, etc. Z-post bénéficie également de la conception modulaire de la suite Z-set permettant à l'utilisateur de définir ses propres procédures de post-traitement par le biais de l'implémentation d'un plugin C++ ou bien d'une fonction Python.

## 5 Calcul performant

Si la suite est interfacée avec les différents solveurs directs de référence (Mumps, Pardiso, Pastix), elle intègre également des solveurs parallèles spécifiques qui permettent d'exploiter pleinement les architectures de calculs massivement parallèles, en tirant aussi parti des connaissances du problème mécanique pour proposer des stratégies dédiées et performantes. Deux stratégies sont proposées : de type décomposition de domaine (5.1) et de type multigrille (5.2). De plus, on tire parti des estimateurs d'erreur *a posteriori* pour proposer des stratégies de calcul réalisant le bon compromis précision / coût de calcul (5.3).

### 5.1 Solveur en décomposition de domaine

Les solveurs en décomposition de domaine consistent à distribuer le maillage global sur différents processeurs, les équilibres locaux sont résolus avec des solveurs directs et un solveur itératif cherche à

obtenir l'équilibre de l'interface entre sous-domaines. Les méthodes classiques BDD et FETI sont disponibles ainsi que la méthode Adaptive Multipreconditionned FETI [6]. Cette dernière permet de résoudre, sur plusieurs milliers de cœurs, des systèmes très mal conditionnés caractéristiques des cas industriels [7], par exemple des bi-matériaux à fort contraste entre les phases, ou des géométries industrielles (aubes, chambres de combustion) résolues avec des modèles matériaux fortement non-linéaires.

## 5.2 Solveurs itératifs et préconditionneur grille grossière

La particularité dans Zébulon est que l'on peut coupler les solveurs itératifs (Gradient conjugué et GMRes) à un préconditionneur, appelé grille grossière [8]. Ce préconditionnement permet de conserver de bonnes propriétés de convergences des solveurs itératifs sur le cas de problèmes fortement non-linéaires. Ce préconditionneur repose sur l'introduction d'un maillage grossier du problème permettant de formuler une matrice de préconditionnement à deux échelles.

Considérons par exemple le cas d'une éprouvette de traction bi-axiale avec un comportement élasto-visco-plastique à deux potentiels. Le problème est résolu en 14 incréments de chargement. On présente dans le tableau 1 les performances obtenues pour différentes discrétisations spatiales.

	654 396 ddl		2 870 580 ddl		4 346 517 ddl	
	Temps (min)	Mém. (Go)	Temps (min)	Mém. (Go)	Temps (min)	Mém. (Go)
MUMPS	18	12	283	96	477	159
CG+grossier	24	4.5	179	20	281	26
Gains	0.75	2.7	1.58	4.8	1.7	6.12

TABLE 1 – Comparaison des solveurs direct (MUMPS [9]) et gradient conjugué préconditionneur grille grossière (CG+grossier - 10 743 ddl) de Z-set sur un calcul élasto-visco-plastique en utilisant 4 cœurs.

D'après les résultats du tableau 1, le solveur itératif combiné au préconditionneur grille grossière se positionne comme une solution viable, pour le calcul non-linéaire à grand nombre de degrés de liberté, sur des infrastructure de calculs limitées.

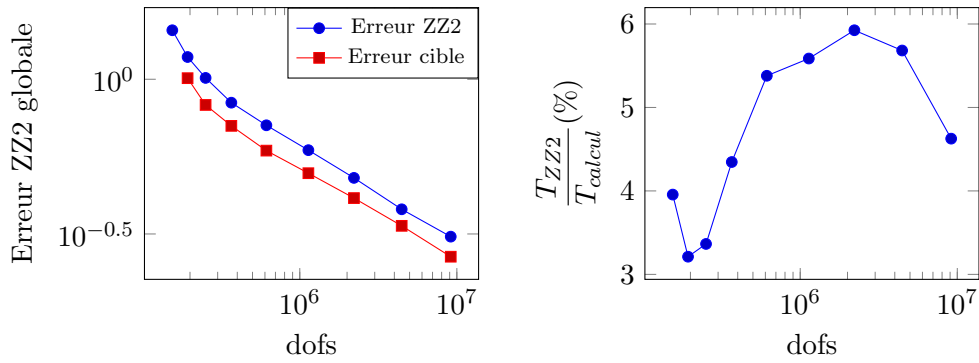
Il se montre notamment économe en mémoire sur toutes les gammes de problèmes, et plus rapide lorsque la taille du problème augmente.

## 5.3 Estimateurs d'erreur et adaptation de maillages

Afin d'atteindre le meilleur compromis précision/coût possible, le solveur Zébulon dispose d'outils d'estimation d'erreur qui s'appuient sur des techniques de lissage [10]. L'implémentation parallèle réalisée tire pleinement partie de l'aspect par patch de la méthode. Ainsi l'estimation d'erreur devient quasiment négligeable sur le coût de calcul global (6% au maximum). L'association de cette indication d'erreur avec un critère d'optimalité du maillage visant à répartir uniformément l'erreur globale sur le nouveau maillage permet de fixer la nouvelle taille de maille de sorte à minimiser le nombre d'éléments. À partir de la carte de taille de maille ainsi définie, les différents outils de remaillage interfacés avec Z-set [11] permettent de réaliser facilement une adaptation.

Le processus d'adaptation de maillage est ici illustré dans deux cas : le calcul élastique linéaire d'une aube de turbine et le calcul en axisymétrique d'une éprouvette cylindrique avec un comportement élasto-plastique endommageable en grandes déformations. Dans le cas de l'aube de turbine (Fig. 3), à chaque itération du processus : (i) résolution du problème avec le maillage  $\mathcal{M}^{(k)}$ ; (ii) estimation d'erreur  $\mathcal{E}_{ZZZ}^{(k)}$  associée à  $\mathcal{M}^{(k)}$ ; (iii) adaptation de  $\mathcal{M}^{(k)}$  en  $\mathcal{M}^{(k+1)}$  pour une erreur cible  $\mathcal{E}_{cible}^{(k+1)} = 0.7 \times \mathcal{E}_{ZZZ}^{(k)}$ <sup>1</sup>. Dans le cas de l'éprouvette cylindrique, l'estimation d'erreur s'appuie sur le champ de plasticité cumulée (voir figure 4) et l'adaptation est faite à partir du maillage en configuration déformée afin de préserver un bon facteur de forme des éléments, donc une bonne précision du calcul.

1. On a volontairement choisi dans cette analyse de ne pas se fixer une erreur cible constante afin de pousser les outils de remaillage et d'adaptation dans leurs retranchements.



(a) Évolution de l'erreur estimée ZZ2 et de l'erreur cible en fonction du nombre de degrés de liberté. (b) Évolution du pourcentage du temps de calcul dédié à l'estimation d'erreur en fonction du nombre de degrés de liberté.

FIGURE 3 – Illustration des performances de l'estimateur d'erreur ZZ2 implémenté dans Z-set sur l'adaptation d'un calcul d'aube élastique linéaire.

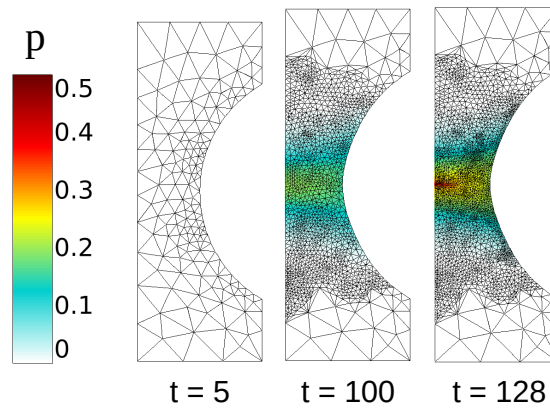


FIGURE 4 – Adaptation de maillage sur la base d'une estimation d'erreur pour un calcul axisymétrique d'une éprouvette cylindrique : le maillage est bien raffiné dans les zones où la plasticité cumulée  $p$  se développe.

#### 5.4 Méthodes de réduction de modèle

Il est possible de mettre en oeuvre la méthode de d'hyper-réduction de modèle [12, 13] dans Z-set. Les bases d'ordre réduit sont construites en post-traitant les calculs d'apprentissage potentiellement effectués sur des codes commerciaux couplé avec Z-post. Ces bases peuvent être construites sur toutes les variables nodales et intégrées stockées. Ensuite, le domaine d'intégration réduit est construit algorithmiquement et un calcul hyper-réduit peut être lancé. Z-set propose aussi des fonctionnalités de reconstruction de la solution hyper-réduite sur le domaine complet par Gappy POD et de calcul d'erreur par rapport à une solution de référence éléments finis.

## 6 Interface Python

Afin d'ouvrir de nouvelles possibilités d'interfaçage de Z-set avec d'autres logiciels, une interface pour Python3 est proposée. Celle-ci répond aux objectifs suivants : permettre à l'utilisateur d'enrichir ses simulations de sous-routines spécialisées, tout en travaillant autant que possible sur les structures de données natives du code ; ouvrir le code à une interaction avec l'écosystème de calcul scientifique de Python ; et enfin faciliter le couplage entre différents codes de calculs, de manière non intrusive.

Cette interface propose une interface simple et intuitive pour manipuler et assembler des composants de résolution d'un problème éléments finis. Elle n'expose que des objets « macroscopiques » pour ne pas nuire aux performances, en utilisant des vues `numpy` des champs de données pour en faciliter la

manipulation.

Deux modes de fonctionnement existent :

- Z-set embarque un interpréteur Python et l'utilisateur peut intervenir à certains points clés du déroulement du calcul (initialisation, début ou fin d'incrément, etc.) pour par exemple modifier à la volée les valeurs d'une condition aux limites ou d'un champ de paramètres, adapter le pas de temps, exporter/importer des données pour coupler deux simulations, etc.
- un script Python importe le module `zset` ; celui-ci comporte des classes et méthodes permettant de manipuler les objets tels que le maillage, la matrice du problème, les champs nodaux et de variables internes, de faire du post-traitement. Il permet d'aller jusqu'à l'écriture complète d'un schéma de résolution d'un problème éléments finis, en combinant les briques macroscopiques telles que l'assemblage et la factorisation des matrices globales du problème (matrice de rigidité et de masse), la boucle en temps, les méthodes de points fixes, etc.

## Exemples d'applications

L'interface Python peut être utilisée pour traiter des problèmes très variés. On peut citer des applications de cosimulation implicite/explicite pour lesquelles les algorithmes de couplage ont été scriptés via cette interface [14, 15]. Un schéma de résolution parallèle en temps a également été mis en œuvre [16], pour lequel les modules `numpy`, `zset` et `MPI` ont été mis en œuvre simultanément.

On peut également réaliser du couplage de codes pour des applications multiphysiques (couplage électro-thermo-mécanique pour des applications d'impact de foudre [17]) ou pour des schémas de résolutions particuliers comme la résolution alternée par point fixe d'une formulation phase field [18].

Dans le cadre du dialogue essai-calcul, l'assemblage de plusieurs composants Python spécialisés s'est aussi montré efficace à la fois comme langage de développement pour le développement d'une plateforme logicielle et comme interface d'utilisation [19] pour l'utilisation de celle-ci.

Le module Python est également très prometteur dans un cadre pédagogique : on peut en effet en quelques dizaines de lignes à la fois synthétiques et compréhensibles, écrire un algorithme de Newton-Raphson et en extraire des informations de convergence pertinentes à la compréhension de la méthode. Ou encore écrire les phases *offline* et *online* d'une stratégie de réduction de modèle de type POD-Galerkin (Fig. 5), pour laquelle on tire parti de la richesse des modules d'algèbre linéaire de `numpy` et `scipy`.

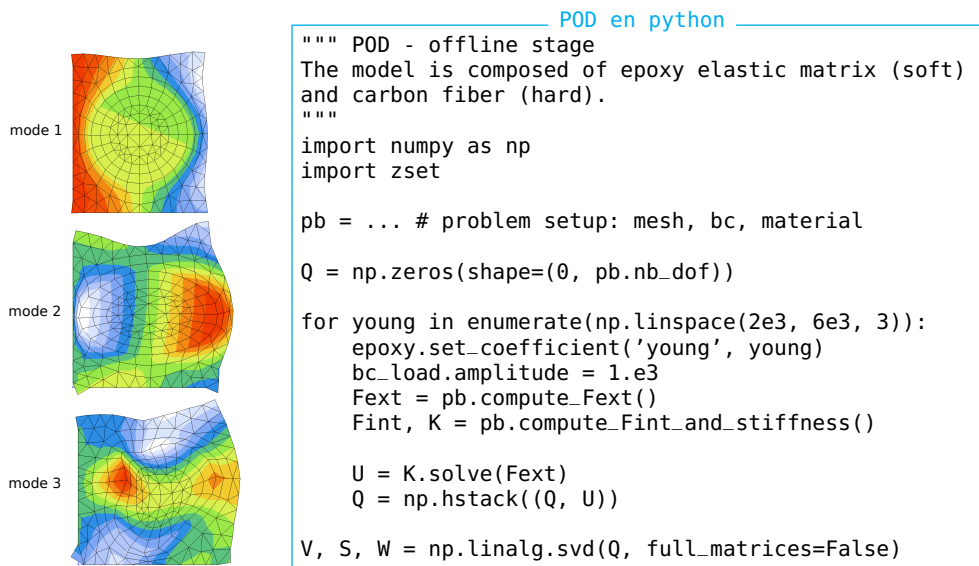


FIGURE 5 – Exemple de mise en œuvre pour la réalisation d'une réduction de modèles.

## 7 Simulation du contact

La suite Z-set dispose de divers algorithmes pour traiter des problèmes du contact avec et sans frottement en deux et trois dimensions. En plus de la méthode classique de pénalisation, qui est implé-



mentée dans le cadre de discrétisation du type nœud-à-surface, la méthode du Lagrangien augmenté monolithique [20, 21] combinée avec la discrétisation du type surface-à-surface (méthode mortar [23]) est également disponible [22]. L'évaluation des intégrales *mortar* est basée sur le découpage précis des intersections des éléments des deux maillages en contact [24]; cela permet d'assurer la précision irréprochable des contraintes dans l'interface et éviter des oscillations parasites présentes dans le contact du type nœud-à-surface.

Les algorithmes de contact dans Z-set sont équipés d'une technique de détection des contacts basée sur des arbres kD, et ils sont fonctionnels sur architectures parallèles à mémoire distribuée. Le choix du coefficient de pénalisation dans la méthode de pénalisation et celle du Lagrangien augmenté se fait automatiquement à la volée pour chaque "élément de contact" en fonction de la raideur locale; ce choix assure la convergence optimale de la méthode de Newton. En outre, dans la boucle de convergence de Newton nous utilisons une séparation des non-linéarités provenant du comportement des matériaux et celles du contact frottant; cette séparation permet de converger des calculs fortement non linéaires, comme un essai de Charpy ou des problèmes d'auto-contact en grandes transformations. Dans la figure 6, un exemple de traitement du contact dans l'attache du type pied de sapin entre une aube et un disque est présenté.

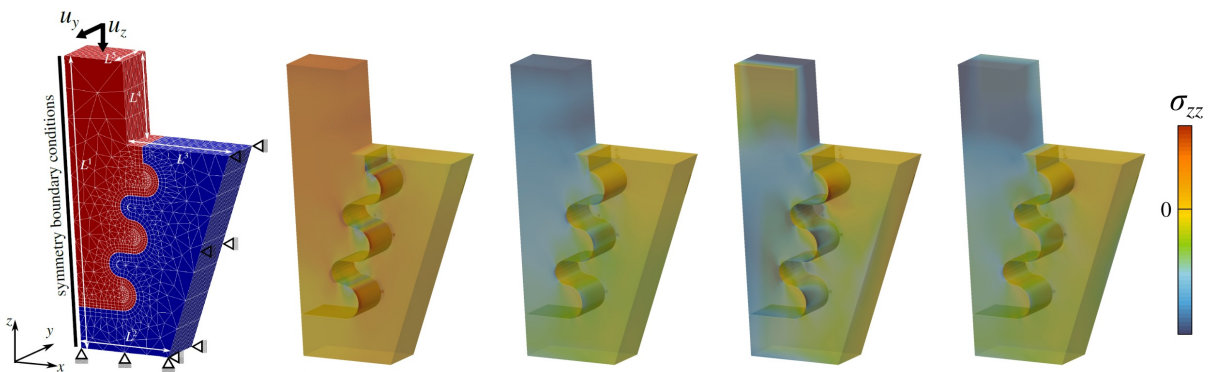


FIGURE 6 – Modèle pied d'aube-disque sous chargement axial et tangentiel : distribution de contrainte axiale aux différents instants de chargement.

## 8 Conclusion

La suite logicielle Z-set comporte de nombreux modules dédiés à la simulation des matériaux et structures. Le module matériau fournit d'une part une base riche de comportements pour les matériaux métalliques (modèles thermo-visco-plastiques, plasticité cristalline, ...), pour les matériaux composites (endommagement, critères de rupture) ou encore pour les polymères. D'autre part, il facilite le développement et l'intégration de nouvelles lois de comportement et peut être relié à certains autres codes de calcul de structures.

Un solveur éléments finis accompagne le module matériau pour fournir une solution complète de simulation en mécanique des structures. Modulaire, ce solveur bénéficie de fonctionnalités récentes en calcul intensif : solveurs linéaires parallèles directs ou itératifs, méthodes de réduction de modèles, adaptation de maillage. Il sert ainsi de plateforme de développement pour des méthodes numériques avancées, tout en permettant leur utilisation et démonstration de performances sur des cas industriels de grande taille et de complexité variée.

Les facilités d'extension du logiciel en font un outil apprécié dans les domaines académiques et industriels, pour tester et développer de nouveaux modèles matériaux ou de nouvelles méthodes numériques (contact, fissuration, HPC, ...). Au-delà de la contribution des co-développeurs principaux (Centre des Matériaux, Onera, Transvalor), l'ouverture à des partenaires extérieurs académiques et industriels permet une extension continue des capacités de la suite Z-set.

## Références

- [1] Z-set – Non-linear material and structure analysis suite, <http://zset-software.com/>
- [2] E. Fessler, E. Andrieu, V. Bonnard, V. Chiaruttini, S. Pierret, *Relation between crack growth behaviour and crack front morphology under hold-time conditions in DA Inconel 718*, International Journal of Fatigue, Volume 96, 2017, Pages 17-27.
- [3] H. Proudhon, J. Li, F. Wang, A. Roos, V. Chiaruttini, S. Forest, *3D simulation of short fatigue crack propagation by finite element crystal plasticity and remeshing*, International Journal of Fatigue, Volume 82, Part 2, 2016, Pages 238-246.
- [4] V. Chiaruttini, D. Geoffroy, V. Riolo, M. Bonnet, *An adaptive algorithm for cohesive zone model and arbitrary crack propagation*, Revue Européenne de Mécanique Numérique/European Journal of Computational Mechanics, Volume 21, 2012, pages 208-218.
- [5] E. Borakiewicz, V. Chiaruttini, A. Hurmane, F. Laurin, *An Adaptive Multi-Model Approach to Simulate the Damage of Composite Panels with Initial Defect under Compressive Loading*, 13th World Congress in Computational Mechanics (WCCM 2018), July 22-27 2018, New-York, USA.
- [6] C. Bovet, A. Parret-Fréaud, N. Spillane, P. Gosselet. *Adaptive multipreconditioned FETI : Scalability results and robustness assessment*, Computers & Structures, 193 :1–20, (2017).
- [7] C. Bovet, A. Parret-Fréaud, P. Gosselet, N. Spillane, *Adaptation à deux niveaux pour le solveur Adaptive Multipreconditioned FETI*, 14<sup>e</sup> colloque national en calcul des structures, Giens, France, May 2019.
- [8] B. Marchand, S. Quilici. *Préconditionnement grille grossière pour la résolution par solveur itératif de problème éléments-finis fortement non-linéaires de grandes tailles*, 14<sup>e</sup> colloque national en calcul des structures, Giens, France, May 2019.
- [9] P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent. *Multifrontal parallel distributed symmetric and unsymmetric solvers*, Computer Methods in Applied Mechanics and Engineering, 501-520, 2000.
- [10] Zienkiewicz O.C., Zhu J.Z., *The superconvergent patch recovery and a posteriori error estimates. Part 1 : The recovery technique*, Int. J. Num. Meth. Eng., 33, pp 1331-1364, 1992
- [11] C. Dapogny, C. Dobrzynski, P. Frey, *Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems* JCP, 262, pp. 358–378, 2014.
- [12] D. Ryckelynck *A priori hyperreduction method : an adaptive approach*, International Journal of Computational Physics, 202 (2005) 346–366.
- [13] D. Ryckelynck *Hyper-reduction of mechanical models involving internal variables*, Int. J. Num. Meth. Eng., 77 (2009).
- [14] T. Chantrait, J. Rannou, A. Gravouil. *Low intrusive coupling of implicit and explicit time integration schemes for structural dynamics : Application to low energy impacts on composite structures*, Finite Elements in Analysis and Design, Elsevier, 2014, 86, 23-33.
- [15] T. Chantrait, J. Rannou, *Couplage implicite/explicite en présence de grand déplacement. Application aux structures tournantes*, 14<sup>e</sup> colloque national en calcul des structures, Giens, France, May 2019.
- [16] M. Gander, L. Halpern, J. Rannou, J. Ryan. *A Direct Time Parallel Solver by Diagonalization for the Wave Equation*, SIAM Journal on Scientific Computing, 2019, 41, A220-A245.
- [17] J. Rannou, C. Huchette, A. Hurmane, F. Tholin. *An analysis of lightning direct effects on CFRP panels. Part II, the electro-thermo-mechanical simulation strategy*, International Conference on Lightning & Static Electricity (ICOLSE 2017), 2017.
- [18] J. Rannou, V. Médeau, F. Laurin, A. Hurmane, F. Lachaud, S. Mousillat, *Rupture de composites tissés : effets d'échelle et capacité des modèles d'endommagement et des techniques de régularisation à les représenter*, 14<sup>e</sup> colloque national en calcul des structures, Giens, France, May 2019.
- [19] JD. Garaud, S. Feld-Payet, F. Bettonte, A. Tireira, Y. Le Sant, G. Le Besnerais, S. Belon. *escale : plateforme logicielle pour le dialogue essai – calcul*, CSMA 2017, Giens.
- [20] P. Alart and A. Curnier. *A mixed formulation for frictional contact problems prone to Newton like solution methods*, Computer methods in applied mechanics and engineering, 92(3) :353–375 (1991).
- [21] V.A. Yastrebov. *Numerical Methods in Contact Mechanics* Wiley/ISTE (2013).
- [22] B.R. Akula. *Extended mortar method for contact and mesh-tying applications* Thèse de doctorat. MINES ParisTech (2019).
- [23] A. Popp, M.W. Gee, and W.A. Wall. *A finite deformation mortar contact formulation using a primal-dual active set strategy*. International Journal for Numerical Methods in Engineering, 79(11) :1354–1391 (2009).
- [24] P. Farah, A. Popp, and W.A. Wall, *Segment-based vs. element-based integration for mortar methods in computational contact mechanics*. Computational Mechanics, 55(1) :209–228 (2015)