

Code mordicus : Méthodes d'Ordre RéDuIt Conçues pour des Usages induStriels

F. Casenave¹, M. Abbas², G. Ferté², D. Ryckelynck³

¹ SafranTech, Digital Sciences and Technologies Department, Rue des Jeunes Bois, Châteaufort, 78114 Magny-les-Hameaux, France, fabien.casenave@safrangroup.com

² EDF R&D, Département Electrotechnique et Mécanique des Structures, Boulevard Gaspard Monge, 91120 Palaiseau, France, {mickael.abbas, guilhem.ferte}@edf.fr

³ MINES ParisTech, PSL University, Centre des matériaux (CMAT), CNRS UMR 7633, BP 87, 91003 Evry, France, david.ryckelynck@minesparis.psl.eu

Résumé — Mordicus est un code open-source pour la réduction de modèle non-intrusive. Un modèle de données générique a été construit pour satisfaire un nombre important de cas d'usage industriels. Le code n'est pas particularisé pour une physique ou une méthode particulière, et propose des modules permettant de combiner des méthodes et données différentes. Des exemples sont fournis en réduction de modèle physique appliquée à la mécanique des structures non-linéaire et en méta-modélisation par processus gaussiens appliquée à la mécanique des fluides.

Mots clefs — réduction de modèle, non-intrusivité, open source.

1. Projet MOR_DICUS

Le projet MOR_DICUS est mené depuis 2019 par un consortium composé d'EDF, Safran, Cemosis, Armines, Sorbonne Université, ESI group, CADLM/Hexagon, Phimeca, Transvalor et CT Ingénierie dans le cadre de l'appel à projet FUI 25 (fonds unique interministériel).

La simulation numérique est de plus en plus utilisée dans un contexte de multiples évaluations : pour une représentation numérique d'un certain objet d'étude, on souhaite évaluer le comportement sur des configurations de sollicitation ou des hypothèses variables. Typiquement, ces variantes peuvent consister en une différence de signaux d'entrée, de paramètres réels du modèle (on parle alors d'étude paramétrique) ou même de géométrie.

La réduction de modèles est une famille d'algorithmes permettant de réduire drastiquement le coût marginal associé à une évaluation, en tirant parti de l'information des évaluations précédentes. En pratique, elle repose sur des techniques d'apprentissage des solutions déjà calculées afin de proposer des approximations de rang faible des opérateurs et de l'espace de définition de la solution.

Ce faisant, elles rendent accessible l'utilisation de modèles numériques même de coût unitaire élevé dans les contextes de multi-évaluations précédents.

L'objectif principal du projet MOR_DICUS est de déployer des méthodes de réduction de modèle génériques dans des briques logicielles pour permettre la réalisation de calculs performants (accélérés, temps-réel), pour la conception et l'analyse rapide et fiable des systèmes industriels, en prenant en compte tous les paramètres d'une modélisation complexe (géométrie, matériaux, conditions aux limites, environnement, etc.).

Pour ce faire, nous avons développé une bibliothèque logicielle partagée open-source mettant en œuvre ces méthodes de réduction de modèles.

Les méthodes de réduction implémentées sont non-intrusives, dans le sens où elles peuvent être mises en œuvre sans avoir à modifier le code du simulateur produisant les solutions des configurations physiques considérées. Ces méthodes appartiennent principalement à deux classes : (i) les méthodes de régression non-linéaires (ou méta modélisation statistiques), et (ii) les méthodes de réduction de modèle physiques, habituellement intrusive, mais dont un effort d'implémentation particulier a permis de rendre strictement non-intrusives dans la bibliothèque mordicus.

2. Modèle de données et architecture de la bibliothèque mordicus

La bibliothèque se compose d'un noyau, où est implémenté le modèle de donnée, et d'un ensemble de modules reposant sur ce noyau et dans lesquels les méthodes de réduction de modèle sont implémentées.

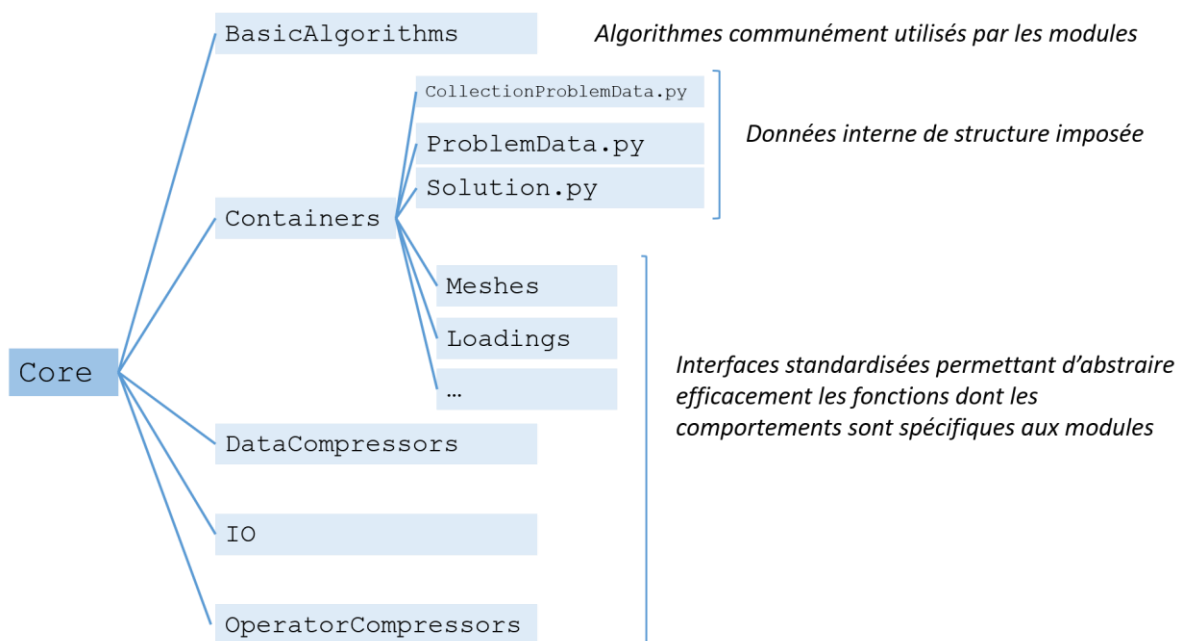


Figure 1 : Architecture du noyau de Mordicus

En règle générale, la mise en œuvre d'une méthode de réduction de modèle se décompose en deux phases :

- Une partie offline, au cours de laquelle des solutions dites « haute-fidélité » sont calculées (à l'aide d'un code de simulation numérique de référence), une méthode de réduction de dimension est utilisée pour compresser les données (DataCompressor, c.f; Figure 1), et des opérations propres à chaque méthode sont exécutées, dont le but est de permettre une construction efficace (rapide) des modèles réduits (OperatorCompressor, c.f; Figure 1). Par exemple, OperatorCompressor correspond à l'entraînement du régresseur dans les méthodes de métamodélisation par processus gaussiens.
- Une partie online, au cours de laquelle le problème réduit est construit et résolu.

L'architecture s'efforce d'être simple et évolutive, avec seulement trois classes imposées dans le modèle de données, et un ensemble de classes pouvant être adaptées à des besoins futurs. Ces trois classes imposées sont `Solution`, `ProblemData` et `CollectionProblemData` (c.f. Figure 1):

- `Solution` contient les snapshots (champs haute-fidélité à chaque piquet de temps pour une quantité donnée), et leur représentation réduite si disponible.
- `ProblemData` décrit un problème physique, contenant le cas échant les valeurs paramétriques de la simulation, des objets `Solution`, conditions aux limites et chargements (Loadings), conditions initiales, lois de comportement...
- `CollectionProblemData` contient l'ensemble des données requises pour l'étude de réduction de modèle considérée : des objets `ProblemData`, des bases d'ordre réduit, ainsi que des données propres aux méthodes de `DataCompressor` et `OperatorCompressor` considérées.

Dans des cas d'usage plus évolués, la frontière entre phases offline et online est plus floue, par exemple lorsque la représentation réduite (base POD par exemple) est enrichie en cours d'exploitation du modèle réduit. La bibliothèque `mordicus` permet de tels cas d'usage. L'utilisation de `mordicus` consiste à peupler la structure de donnée `CollectionProblemData`, puis à exécuter des méthodes numériques qui vont mettre à jour les éléments dans cette structure de données. Ainsi, tant que l'algorithme fait sens, les briques élémentaires associées à chaque méthode numérique peuvent être exécutées dans n'importe quel ordre, et produire des workflows originaux.

3. Deux cas d'usage

3.1. POD-ECM en mécanique des structures non-linéaire

Un module dédié à la mécanique des structures pour les matériaux non-linéaires est disponible dans `mordicus`. Il met en œuvre `DataCompression` par snapshots POD (Proper Orthogonal Decomposition [1]), et `OperatorCompression` par ECM (Empirical Cubature Method [2]). Cette méthode est très proche d'un travail proposé plus tôt, avec une légère variante du calcul de la quadrature réduite [3]. Ce module correspond à la ré-implémentation des travaux [4] dans la bibliothèque `mordicus`. Il a également été appliqué dans [5], où une étude de propagation d'incertitude a été réalisée en appliquant des estimateurs de Monte-Carlo calculés avec un modèle réduit. Dans un contexte de calcul de durée de vie d'aube de turbine, il s'agit de quantifier l'incertitude de champs d'endommagement générée par l'incertitude sur le champ de chargement thermique en entrée d'un calcul elastoviscoplastique, c.f. Figure 2.

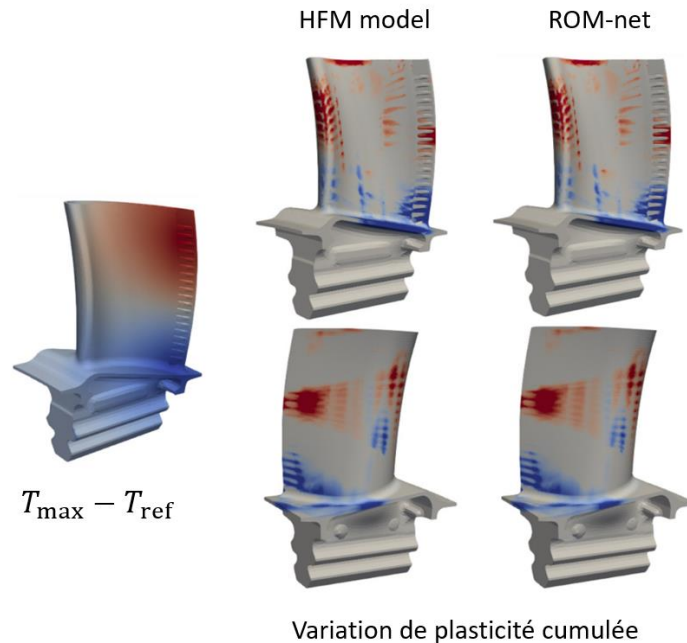


Figure 2 – Comparaison entre modèle haute-fidélité de référence et la prédiction du modèle réduit sur la variation de plasticité cumulée générée par une variation donnée du champ de température

L'utilisation de la bibliothèque mordicus se fait de cette façon :

- Phase offline :
 - Des solutions de déplacement associées à 24 simulations haute-fidélité pour 24 tirages de la température sont lues et peuplent 24 objets `Solution`, associés à 24 objets `ProblemData`, eux-mêmes associés à un `CollectionProblemData`.
 - Une fonction `SnapshotsPOD` est appliquée à `CollectionProblemData` pour construire des bases POD associées au déplacement et aux quantités d'intérêt.
 - Une fonction `CompressOperator` est appliquée à `CollectionProblemData` pour calculer une formule de quadrature réduite associée à l'assemblage de vecteur des forces internes, et pour entraîner un algorithme de reconstruction de quantités duales (par exemple Gappy-POD).
 - Comme la seule variabilité du problème est la température, il est possible de précalculer et préassembler un `ProblemData` associé au modèle réduit (qui sera exécuté dans un second temps) en lui associant lois de comportement et en compressant sur la base POD condition initiale, et chargements centrifuge et pression.
- Phase online :
 - La phase online consiste à tirer des valeurs de la température comme réalisations d'un processus aléatoire, puis les affecter au `ProblemData` préconstruit, pour ensuite évaluer le modèle réduit. La simulation rapide de nombreux calculs réduits permet de calculer des estimateurs Monte-Carlo des quantités d'intérêt en temps raisonnable.

Sur cette application, le facteur d'accélération du modèle réduit par rapport à un modèle haute-fidélité performant, parallèle (décomposition de domaine) est de 636, pour une erreur de l'ordre de 1 à 3% sur les quantités d'intérêt.

3.2. Régression nonlinéaire par processus gaussiens en mécanique des fluides

Une méthode de régression par processus gaussiens [6] est implémentée dans le noyau de la bibliothèque mordicus. L'application est le calcul de l'écoulement autour des aubes d'un compresseur basse pression. La variabilité du problème comprend 38 paramètres physiques et géométriques, et la quantité d'intérêt est la vue méridienne du champ de Mach, c.f. Figure 3.

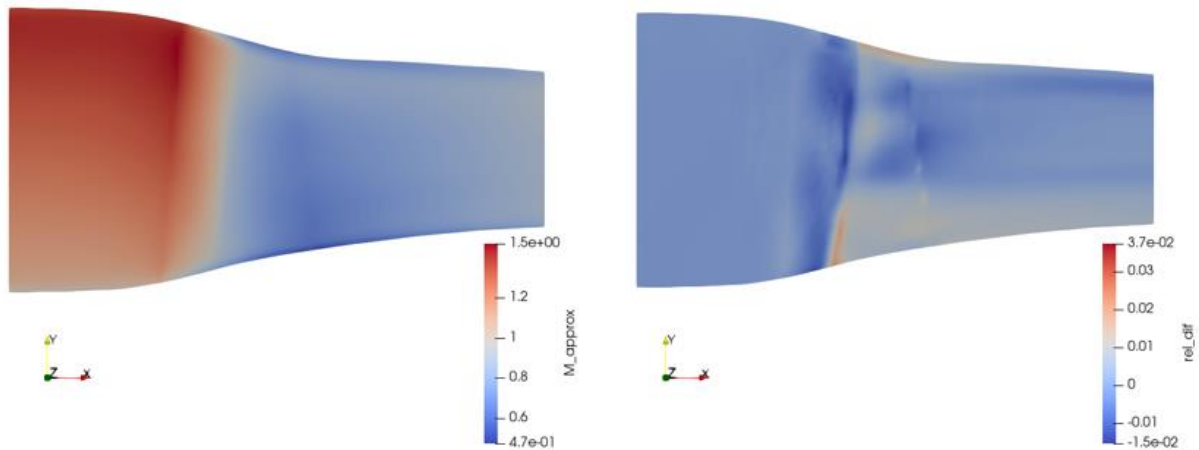


Figure 3 – Champ de Mach sur la vue méridienne d'un compresseur basse pression : (gauche) champ approché par réduction de modèle, (droite) erreur relative en référence et modèle réduit

L'utilisation de la bibliothèque mordicus se fait de cette façon :

- Phase offline :
 - Des solutions de champs de Mach associés à 1000 simulations d'un ensemble d'entraînement sont lues et peuplent 1000 objets Solution, associés à 1000 objets ProblemData, eux-mêmes associés à un CollectionProblemData.
 - Une fonction SnapshotsPOD est appliquée à CollectionProblemData pour construire des bases POD associées au champ de Mach.
 - Une fonction CompressOperator est appliquée à CollectionProblemData entraîner le métamodèle par processus gaussiens.
- Phase online :
 - La phase online consiste à évaluer le modèle réduit sur des nouvelles valeurs des paramètres.

Sur cette application, l'erreur est de l'ordre de 4% sur l'ensemble de test.

Remerciements

Les auteurs remercient Bpifrance, la Région Ile-de-France et la région Grand-Est pour le soutien financier apporté au projet MOR_DICUS qui a permis la réalisation de ce travail.

Références

- [1] L. Sirovich. Turbulence and the dynamics of coherent structures, parts I, II, and III. Q Appl Math., 561-590, 1987
- [2] J. A. Hernández, M. A. Caicedo, A. Ferrer. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. Comput Methods Appl Mech Eng. 313, 687-722, 2017

- [3] C. Farhat, P. Avery, T. Chapman, J. Cortial. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *Int J Numer Methods Eng.*, 98(9), 625-662, 2014
- [4] F. Casenave, N. Akkari, F. Bordeu, C. Rey and D. Ryckelynck, A Nonintrusive Distributed Reduced Order Modeling Framework for nonlinear structural mechanics – application to elastoviscoplastic computations, *Int. J. Numer. Meth. Eng.* 121, 32–53, 2020
- [5] T. Daniel, F. Casenave, N. Akkari, D. Ryckelynck, C. Rey. Uncertainty quantification for industrial numerical simulation using dictionaries of reduced order models, accepted in *Mechanics & Industry* on 22/12/2021.
- [6] C. E. Rasmussen, C. K. I. Williams. *Gaussian Processes for Machine Learning*, the MIT Press, 2006