

GEMSEO[®], une librairie pour l'optimisation multidisciplinaire

F. Gallard¹, S. Béchet¹, S. Bocquet¹, F. Castaneda¹, L. Cousin¹, A. Dechaume¹, R. Espoey¹, J.-F. Figué¹, A. Gazaix¹, J.-C. Giret¹, C. Labouffie¹, M. De Lozzo¹, O. Sapin¹, N. Roussouly¹, G. Ruiz Jimenez¹, A. Scotto di Perrotolo¹, C. Thoulon¹

¹ IRT Saint Exupéry, Toulouse

Résumé — GEMSEO[®] est une librairie *open source* de calcul scientifique en Python permettant d'automatiser la création de processus de simulation et d'optimisation. Elle met à disposition un ensemble d'algorithmes dédiés à la planification d'expériences, optimisation, couplage, régression, quantification d'incertitudes, calcul de dérivées couplées par méthode adjointe ou différenciation automatique, modèles de substitution ou la visualisation de données. Ces algorithmes sont interopérables. GEMSEO s'accompagne de nombreuses extensions, permettant notamment d'accéder à des ressources distantes (e.g. SSH, ordonnancement de tâches sur des plateformes HPC).

Mots clés — simulation numérique, optimisation multidisciplinaire, MDO, UQ, HPC

1 Introduction

Présentation du logiciel et applications

GEMSEO[®] [1] est une librairie de calcul scientifique qui permet la création automatique de processus d'optimisation multidisciplinaire (MDO - *multidisciplinary design optimization*). Elle est utilisée dans le domaine aéronautique et spatial pour l'optimisation d'ailes d'avions [2], d'intégration propulsive [3], de dimensionnement structural [4], de déploiement de flotte d'avions afin de minimiser l'impact environnemental [5], ou de sondes spatiales en présence d'incertitudes [6].

L'optimisation multidisciplinaire

La programmation d'un processus d'optimisation automatique nécessite de coordonner l'exécution d'un ensemble de logiciels de simulation, appelés "disciplines", et de gérer les échanges de données entre ces logiciels. Dans le cas de la MDO, ces logiciels peuvent être nombreux, tout comme les variables d'interaction qui les lient, appelées "variables de couplage". Ces variables de couplage peuvent engendrer une inter-dépendance entre un ensemble de disciplines. Ces dernières sont alors dites fortement couplées. Par exemple, dans le second graphe de couplage de la figure 1, la discipline "1" prend "x" en entrée et retourne "y" en sortie, tandis que la discipline "2" prend "y" en entrée, et retourne "x" en sortie. Un couplage faible est illustré sur le premier graphe de couplage de cette même figure. Pour résoudre les couplages forts, il existe un ensemble d'algorithmes de MDA (*multidisciplinary design analysis*) issus notamment des méthodes de décomposition de domaine, tels que les méthodes de Gauss-Seidel et Jacobi non-linéaires par blocs. En MDO, un algorithme itératif pilote l'ensemble du processus d'optimisation. Une autre stratégie de résolution des couplages forts consiste alors à ajouter des copies des variables de couplage fort dans l'espace de recherche du problème d'optimisation, ainsi que des contraintes d'égalité. Ces dernières sont dites de consistance car elles assurent qu'à convergence, les copies sont égales aux variables de couplage fort en sortie des disciplines. Cette stratégie de formulation MDO est appelée IDF (*individual design feasible*); elle est une alternative à la formulation MDF (*multidisciplinary design feasible*) qui s'appuie sur une MDA à chaque itération de l'algorithme d'optimisation pour éliminer les variables de couplage fort.

2 Automatisation du processus

Le concept clé de GEMSEO est l'automatisation de la génération du processus MDO à partir d'une formulation MDO. L'utilisateur commence par déclarer ses disciplines via leurs interfaces de programmation ou en appelant des exécutables. Chaque discipline est alors définie par ses variables d'entrée et de sortie. Ensuite, il choisit la formulation MDO qu'il souhaite utiliser. De là, GEMSEO génère automatiquement le processus de calcul associé. Cette automatisation repose sur la consistance de la définition et du nommage des variables entre les disciplines. La figure 1 illustre le graphe de couplage et le processus de calcul ainsi générés à partir de la déclaration de deux disciplines faiblement ou fortement couplées. Le processus est représenté sous la forme d'un diagramme XDSM, qui est un standard de représentation dédié à la MDO [7].

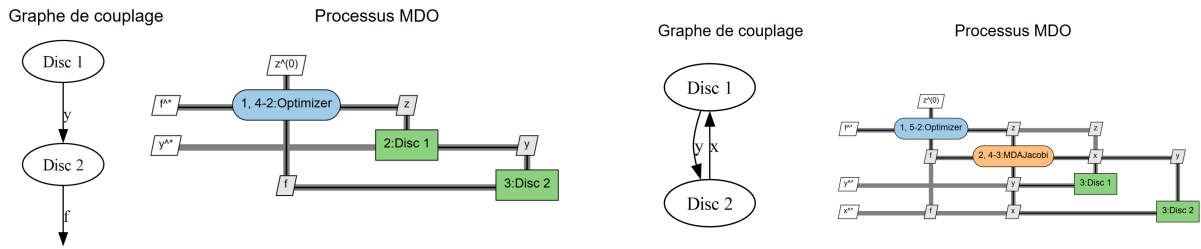


FIGURE 1 – Génération automatique du processus de calcul en fonction de la nature du couplage (faible à gauche, fort à droite).

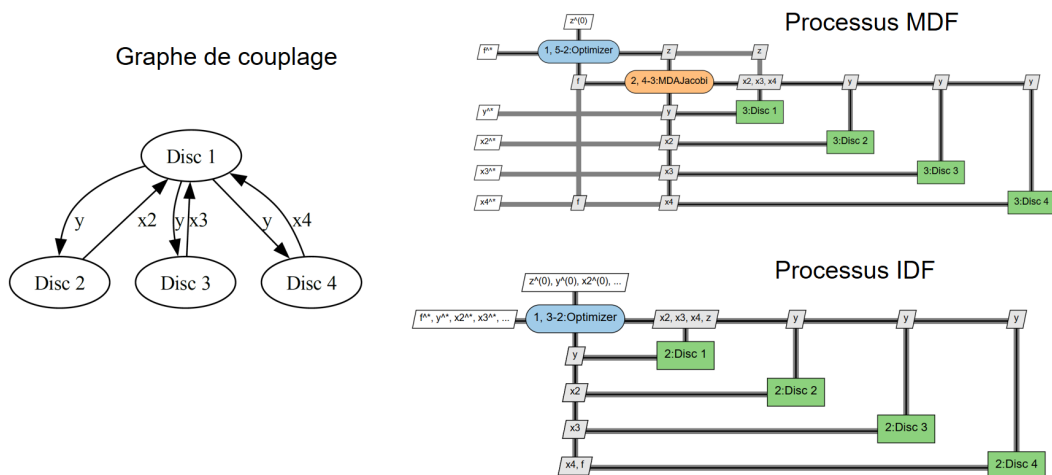


FIGURE 2 – Génération automatique du processus de calcul en fonction de la formulation MDO (MDF en haut, IDF en bas).

La figure 2 montre l'intérêt de cette génération pour la reconfiguration du processus MDO où IDF et MDF sont interverties. GEMSEO ne nécessite pas de développement supplémentaire de la part de l'utilisateur pour changer de formulation. Cela permet également de valider le code du processus sur des cas de référence maîtrisés, ce qui est complexe pour des cas réels. Il est nécessaire de disposer d'alternatives en la matière car il n'existe pas une formulation MDO unique qui soit la meilleure quelque soit le problème et notamment la dimension de ses variables de couplage [8].

La figure 3 montre l'intérêt de cette génération quand un grand nombre de disciplines est impliqué. GEMSEO est capable de détecter l'ensemble des cycles de couplage fort dans le graphe de couplage, de générer un sous-ensemble minimal de MDA et de les paralléliser automatiquement. Cela est utilisé à grande échelle avec parfois des milliers de disciplines. La programmation manuelle de telles MDA sur des problèmes de cette taille rend la maintenance très difficile et montre donc la valeur ajoutée importante de GEMSEO sur ce point. D'autre part, GEMSEO est capable de calculer automatiquement la dérivée totale des sorties des processus, et notamment des MDA, par méthode adjointe couplée proposée

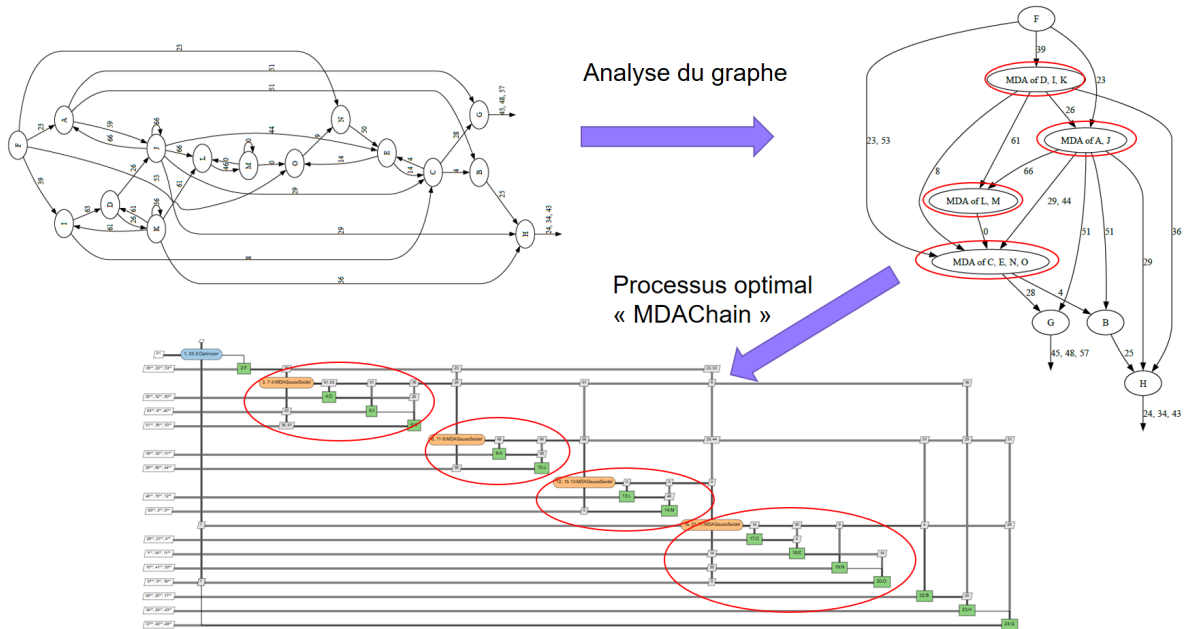


FIGURE 3 – Génération automatisée de sous-blocs de MDA à partir du calcul de graphe (algorithme de Tarjan).

initialement dans [9] et appliquée classiquement pour les problèmes couplés aéro-structure [10].

3 Aperçu des fonctionnalités

GEMSEO offre différents dispositifs de persistance des données. Tout d’abord, les évaluations de l’objectif et des contraintes sont sauvegardées dans une base de données au fil de l’optimisation. Ceci permet de relancer l’optimisation à partir de la dernière itération en cas de souci technique conduisant à un arrêt prématuré. Cela permet aussi ne pas réévaluer les fonctions en un point présent dans l’historique d’optimisation. Par ailleurs, les disciplines disposent elles aussi d’un mécanisme de cache, permettant de sauvegarder les différentes évaluations afin d’analyser plus finement l’état multidisciplinaire au cours de l’optimisation. Enfin, ces deux types de base de données peuvent être exportés dans un standard tabulaire facilitant l’analyse de données *a posteriori* (statistiques, visualisation, etc.).

Par ailleurs, GEMSEO donne accès à un large catalogue d’algorithmes interopérables, dont nous donnons un bref aperçu ci-dessous. Certains algorithmes sont des interfaces vers des bibliothèques spécialisées tandis que d’autres ont été implémentées directement dans GEMSEO.

Planification d’expériences Des algorithmes de plan d’expériences (*Design of Experiment - DoE*) des bibliothèques OpenTURNS [11] et pyDOE sont interfacés, permettant l’exploration de design, l’analyse de compromis multi-disciplinaires (*trade-off*), mais aussi la construction d’échantillons statistiques pour l’analyse de sensibilité ou la construction de modèles de substitution.

Optimisation Plusieurs interfaces vers des bibliothèques *open-sources* d’optimisation sont disponibles donnant accès à plusieurs familles d’algorithmes : global ou local, avec ou sans gradient, génétique ou déterministe, ou bien basé sur la construction d’un modèle de substitution. Citons par exemple la bibliothèque Python SciPy avec des algorithmes locaux basés gradient tels que SLSQP ou BFGS, des algorithmes globaux évolutionnaires, ou bien des algorithmes sans gradient tels que NELDER-MEAD ou COBYQA [12], [13]. La famille des algorithmes sans gradient de Powell (COBYLA, BOBYQA...) sont disponibles via des interfaces aux bibliothèques PDFO [14] et NLOPT [15]. Une interface à pyMOO [16] donne accès à plusieurs algorithmes génétiques tel que NSGA2. L’algorithme global EGO (*Efficient Global Optimization*) basé sur des modèles de substitution du type processus Gaussien est également accessible avec une interface à la bibliothèque SMT (*Surrogate Modeling Toolbox*) [17]. Certains algorithmes propriétaires,

tel que SNOPT [18] interfacé via PyOptSparse [19], sont également disponibles. D'autres, comme le Lagrangien augmenté, ont été développés en interne, tout comme des stratégies multi-objectifs telle que la méthode mNBI [20].

MDA Ces algorithmes permettent de résoudre les systèmes d'équations non-linéaires pour le calcul de couplages forts entre disciplines via deux approches : les méthodes de point fixe avec les algorithmes non-linéaires par blocs Jacobi et Gauss-Seidel (avec des méthodes d'accélération), et les méthodes de recherche de zéro tel que Newton et quasi-Newton. Les MDA peuvent être automatiquement construites par analyse du graphe de couplage entre les disciplines et plusieurs algorithmes MDA peuvent être composées séquentiellement.

Formulation MDO GEMSEO implémente les formulations monolithiques MDF et IDF décrites précédemment qui permettent de résoudre les couplages soit via la résolution d'une MDA, soit en introduisant des variables additionnelles et des contraintes de consistance à l'optimiseur. Des formulations distribuées, bi-niveaux, sont également implémentées sous le nom de formulation *bi-level* [21] et *bi-level BCD (Block Coordinate Descent)* [22], [2], permettant de décomposer le problème d'optimisation initial en sous-problèmes. De manière plus générale, GEMSEO possède les briques élémentaires et les mécanismes permettant d'implémenter simplement de nouvelles formulations distribuées.

Visualisation De nombreux post-traitements sont disponibles pour la visualisation des résultats d'optimisation tels que les historiques de convergence (variables, objectif, contraintes), des *radar charts*, des *parallel coordinates*, mais aussi pour l'analyse de sensibilité (*scatter plot*, *gradient sensitivity*) ou l'exploration d'espace avec les *Self-Organizing Map (SOM)*.

Modèles de substitution et apprentissage Les techniques d'apprentissage pour le *clustering*, la classification ou la régression sont disponibles via des interfaces aux bibliothèques OpenTURNS et scikit-learn. Les mesures de qualité classiques sont disponibles (R^2 , MSE) avec des estimations par technique de ré-échantillonnage (*cross-validation* et *bootstrap*). Il est également possible de construire les modèles sur des variables latentes après transformation des données, telle qu'une analyse en composantes principales.

Incertitudes Des incertitudes probabilistes (variables aléatoires) peuvent être prises en compte en entrée des disciplines via l'ajout de distributions de probabilité et la définition d'un espace aléatoire. Cela permet de réaliser des analyses de sensibilité (indices de corrélation, analyse de Sobol, criblage de Morris...), de propager les incertitudes pour estimer des statistiques sur les sorties et d'évaluer la fiabilité par rapport à des critères de défaillance. Les méthodes d'estimation des statistiques (par échantillonnage ou modèle de substitution) peuvent être combinées à la résolution d'un problème MDO pour le résoudre en présence d'incertitudes : on parle d'UMDO [23].

4 MDO distribuée, multi-machine, multi-OS, en entreprise étendue, HPC

Les processus MDO à l'échelle industrielle ou collaboratifs nécessitent souvent plusieurs machines pour s'exécuter. Ce besoin découle de la demande de ressources spécialisées telles que des logiciels, des systèmes d'exploitation, du matériel ou une puissance de calcul accrue. De plus, les collaborations multi-partenaires donnent souvent la priorité à la protection de la propriété intellectuelle en obscurcissant les modèles internes et en n'exposant que les interfaces et les contrats d'interface pour les disciplines. Bien que les technologies telles que les planificateurs de tâches, les clients et serveurs SSH et HTTP et les plateformes *cloud* soient largement disponibles, leur adoption dans les process MDO est complexe techniquement. La mise en œuvre de ces technologies présente des défis importants tant pour les utilisateurs finaux que pour les intégrateurs de processus MDO. Des efforts ont été déployés dans GEMSO pour apporter une gamme de fonctionnalités de plateforme permettant de décharger de manière sélective et transparente certaines parties du processus vers des machines distantes, et donc de déployer la MDO en entreprise étendue :

- une discipline de planification de tâches pour déléguer les calculs au HPC avec des planificateurs de tâches tels que LSF, PBS ou SLURM.
- une discipline SSH qui permet de déléguer le calcul d'une discipline à un nœud distant accessible via le protocole SSH,
- une discipline HTTP qui fournit un client et un serveur permettant d'exposer les disciplines GEMSEO en tant que services web.
- un plugin portant la couche de processus de GEMSEO (disciplines, MDAs, etc.) vers le protocole MPI qui permet d'échanger les données via les réseaux haute performance.

Ces fonctionnalités introduisent de nouveaux défis pour le processus MDO décentralisé, notamment en matière d'observabilité et de fiabilité. En ce qui concerne la fiabilité, compte tenu du risque de pannes matérielles ou d'instabilité du réseau dans le cadre du calcul à distance, des développements spécifiques ont été réalisés pour traiter de manière sélective les erreurs pouvant résulter de problèmes matériels ou réseau et de relancer automatiquement les calculs quand c'est nécessaire. Enfin, des capacités ont été développées pour permettre l'observabilité à travers les composants distants des processus MDO distribués. La figure 4 illustre ces capacités.

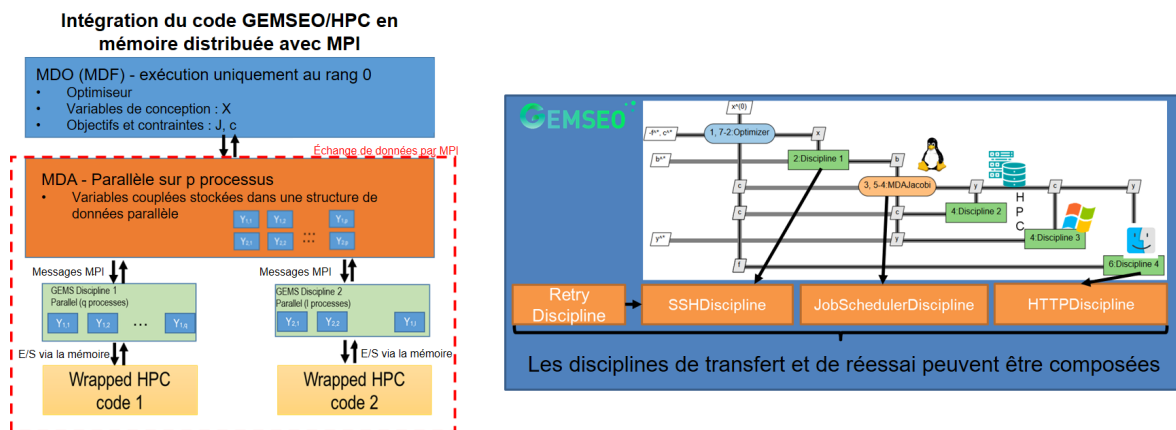


FIGURE 4 – Fonctionnalités de calcul distribué, multi-machine, multi-OS, MPI.

5 Extensions

GEMSEO dispose de nombreuses extensions permettant d'enrichir les fonctionnalités décrites en section 3; elles sont listées dans la figure 5. Ces modules étendent les choix des algorithmes (optimisation, *DoE*...) ou encore la capacité d'exécuter des simulateurs sous forme de disciplines (Nastran, Matlab, Scilab, FMU...). Il existe aussi des extensions permettant de traiter facilement d'autres classes de problèmes, tels que la calibration, ou l'intégration temporelle. Enfin, une librairie Open Source appelée VIMSEO et développée à l'IRT Saint Exupéry étend GEMSEO pour permettre de démontrer la crédibilité des modèles numériques et supporter la Certification par Analyse sur la base de la méthodologie du VV&UQ.

6 Feuille de route

GEMSEO[®] est en constant développement depuis sa création en 2015, sa feuille de route concernant son développement est illustrée dans la figure 6. Les développements dans le cœur MDO se concentrent sur les capacités avancées et leurs synergies telles que l'optimisation combinant les techniques multi-niveau, multi-objectif et multi-fidélité, l'optimisation de problèmes dépendant du temps. Les méthodes gérant les incertitudes couvrent un vaste champ, et doivent se focaliser sur les problématiques de temps de calcul, en synergie notamment avec les modèles de substitution ou les calculs de gradient. Sur le champ applicatif, la demande est forte de traiter des problèmes de choix d'architecture de systèmes complexes, ce qui amène à une structure hiérarchique des espaces de conception et des disciplines. La dimension système de systèmes est également à développer, en lien avec les méthodes avancées d'ingénierie système.

Name / Repo	Description	Documentation
gemseo-benchmark	A GEMSEO-based package to benchmark optimization algorithm.	Documentation
gemseo-calibration	Capability to calibrate GEMSEO disciplines from data	Documentation
gemseo-fmu	GEMSEO plugin for FMU dynamic models	Documentation
gemseo-hexaly	GEMSEO interface to Hexaly solver	Documentation
gemseo-jax	GEMSEO plugin for JAX	Documentation
gemseo-matlab	GEMSEO plugin for MATLAB.	Documentation
gemseo-mllearning	Miscellaneous machine learning capabilities	Documentation
gemseo-mma	GEMSEO plugin for the MMA (Method of Moving Asymptotes) algorithm.	Documentation
gemseo-pdfio	GEMSEO plugin for the PDFIO library.	Documentation
gemseo-petsc	PETSc wrapper for LinearSolver and BaseMDA	Documentation
gemseo-pseven	GEMSEO plugin for the pSeven library.	Documentation
gemseo-pymoo	Pymoo wrapper for optimization algorithms	Documentation
gemseo-pyoptsparse	GEMSEO interface to pyoptsparse algorithms.	Documentation
gemseo-scilab	Interfacing Scilab functions	Documentation
gemseo-ssh	SSH plugin for GEMSEO	Documentation
gemseo-template-editor-gui	A GUI to create input and output file templates for DiscFromExe.	Documentation
gemseo-umdo	Capability for MDO under uncertainty	Documentation

FIGURE 5 – Extensions de GEMSEO.

Ces approches permettent également de mieux définir l'étude MDO [24]. L'interopérabilité de GEMSEO avec d'autres *frameworks* de simulation est en cours de développement, basée sur des initiatives en ce sens [25].

7 Site internet, forum utilisateur, forge

Le site internet du projet : <http://www.gemseo.org>.

Le forum utilisateur : <https://gemseo.discourse.group/>.

La forge logicielle du projet : <https://gitlab.com/gemseo/dev/gemseo>.

Remerciements

Ces travaux ont été menés avec le soutien financier du programme France 2030 dans le cadre du projet ANR-10-AIRT-01. Ils ont aussi été soutenus par le projet NEXTAIR, qui a reçu un financement du programme de recherche et d'innovation Horizon Europe de l'Union européenne dans le cadre de la convention de subvention n° 101056732. Les points de vue et opinions exprimés sont toutefois ceux

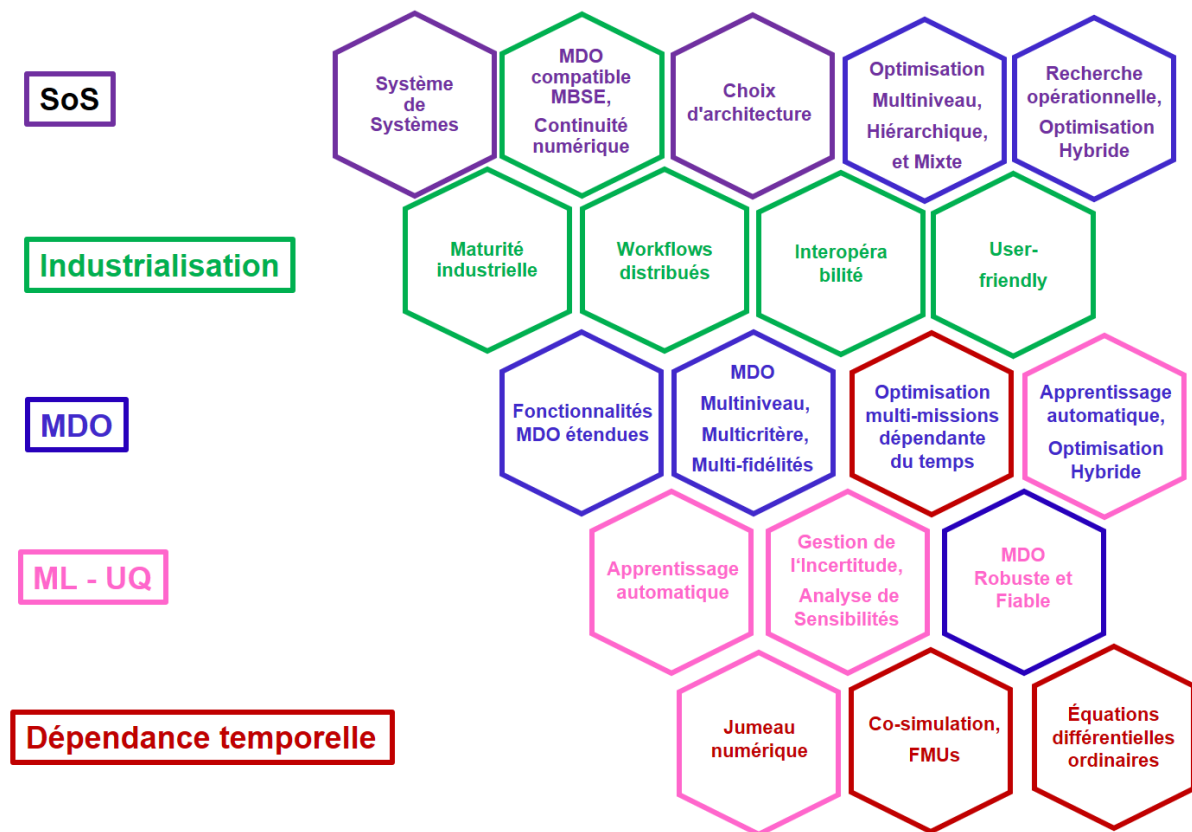


FIGURE 6 – Feuille de route de GEMSEO.

des auteurs uniquement et ne reflètent pas nécessairement ceux de l'Union européenne. Ni l'Union européenne ni l'autorité octroyant la subvention ne peuvent en être tenues responsables.

Références

- [1] François Gallard, Charlie Vanaret, Damien Guénot, Vincent Gachelin, Rémi Lafage, Benoit Pauwels, Pierre-Jean Barjhoux, and Anne Gazaix. GEMS : a Python Library for Automation of Multidisciplinary Design Optimization Process Generation. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0657, 2018.
- [2] Gilberto Ruiz Jimenez, Nicolas Roussouly, François Gallard, Anne Gazaix, Joël Brezillon, Kittinan Thanissaranon, Vida Brück, and Joaquim RRA Martins. A Comparison of MDO Bi-Level Formulation Variants with MDF on a High-Fidelity Aeroelastic Wing Test Case. In *AeroBest 2025*, 2025.
- [3] Anne Gazaix, Nicolas Roussouly, Matthias De Lozzo, François Gallard, Maxime Hamadi, Vincent Ambert, Vincent Drouet, Charles Thoulon, Thierry Druot, Simone Coniglio, et al. High Fidelity Multi-Component Wing, Pylon, Nacelle and Mounts Industrial MDO Application. In *AIAA AVIATION FORUM AND ASCEND 2025*, page 3556, 2025.
- [4] Shouvik Bandopadhyay, François Gallard, and Simone Coniglio. Preconditioning Techniques for General Constrained Optimization Problems : Application to Industrial Aerospace Component Design. In *AeroBest 2025*, 2025.
- [5] Ian Costa-Alves, Nicolas Gourdain, François Gallard, and Anne Gazaix. Optimal aircraft fleet and energy mix under limited availability of resources. In *26th Conference of the International Society for Air Breathing Engines*, 2024.
- [6] Olivier Sapin, Loïc Cousin, Nicolas Roussouly, François Gallard, Anne Gazaix, Matthias De Lozzo, Xavier Fosse, Nicolas Sarda, and Gaspard Berthelin. Industrial Application of a Multi-Disciplinary Design Optimization with Uncertainties to a Pair of Telecommunication Satellites. In *AIAA AVIATION FORUM AND ASCEND 2025*, page 3737, 2025.
- [7] Andrew B Lambe and Joaquim RRA Martins. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 46 :273–284, 2012.

- [8] Charlie Vanaret, François Gallard, and Joaquim RRA Martins. On the Consequences of the "No Free Lunch" Theorem for Optimization on the Choice of an Appropriate MDO Architecture. In *18th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 3148, 2017.
- [9] Jaroslaw Sobieszcanski-Sobieski, Jeremy Agte, and Robert Sandusky, Jr. Bi-Level Integrated System Synthesis (BLISS). In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, page 4916, 1998.
- [10] Joaquim RRA Martins, Juan J Alonso, and James J Reuther. A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design. *Optimization and Engineering*, 6(1) :33–62, 2005.
- [11] Michaël Baudin, Anne Dutfoy, Bertrand Iooss, and Anne-Laure Popelin. *OpenTURNS : An Industrial Software for Uncertainty Quantification in Simulation*, pages 1–38. Springer International Publishing, Cham, 2016.
- [12] T. M. Ragonneau. *Model-Based Derivative-Free Optimization Methods and Software*. PhD thesis, Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China, 2022.
- [13] T. M. Ragonneau and Z. Zhang. COBYQA Version 1.1.3, 2025.
- [14] Tom M Ragonneau and Zaikun Zhang. PDFO : a cross-platform package for Powell’s derivative-free optimization solvers. *Mathematical Programming Computation*, 16(4) :535–559, 2024.
- [15] Steven G Johnson and Julien Schueller. NLOpt : Nonlinear optimization library. *Astrophysics Source Code Library*, pages ascl–2111, 2021.
- [16] Julian Blank and Kalyanmoy Deb. Pymoo : Multi-Objective Optimization in Python. *IEEE Access*, 8 :89497–89509, 2020.
- [17] Paul Saves, Rémi Lafage, Nathalie Bartoli, Youssef Diouane, Jasper Bussemaker, Thierry Lefebvre, John T Hwang, Joseph Morlier, and Joaquim RRA Martins. SMT 2.0 : A Surrogate Modeling Toolbox with a focus on hierarchical and mixed variables Gaussian processes. *Advances in Engineering Software*, 188 :103571, 2024.
- [18] Philip E Gill, Walter Murray, and Michael A Saunders. SNOPT : An SQP algorithm for large-scale constrained optimization. *SIAM review*, 47(1) :99–131, 2005.
- [19] Ella Wu, Gaetan Kenway, Charles A Mader, John Jasa, and Joaquim RRA Martins. pyOptSparse : A Python framework for large-scale constrained nonlinear optimization of sparse systems. *Journal of Open Source Software*, 5(54) :2564, 2020.
- [20] Pradyumn Kumar Shukla. On the normal boundary intersection method for generation of efficient front. In *International Conference on Computational Science*, pages 310–317. Springer, 2007.
- [21] Anne Gazaix, François Gallard, Vincent Ambert, Damien Guénot, Maxime Hamadi, Stéphane Grihon, Patrick Sarouille, Thierry Y Druot, Joël Brézillon, Vincent Gachelin, et al. Industrial Application of an Advanced Bi-level MDO Formulation to Aircraft Engine Pylon Optimization. In *AIAA Aviation 2019 Forum*, page 3109, 2019.
- [22] Yann David. *Bi-level architectures and multi-fidelity algorithms for multidisciplinary optimization in high dimensions*. PhD thesis, Université de Toulouse, 2024.
- [23] M. De Lozzo, C. Labouffie, O. Sapin, N. Roussouly, F. Gallard, A. Aziz-Alaoui, A. Dechaume, and A. Gazaix. Multi-disciplinary design optimization under uncertainty : the open source capabilities of GEMSEO. In *UNCECOMP 2025, 6th ECCOMAS Thematic Conference on Uncertainty Quantification in Computation Sciences and Engineering*, M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.), 2025.
- [24] Anne Gazaix, Vincent Ambert, Mickael Guillen, Nicolas Roussouly, Francois Gallard, Thierry Druot, Simone Coniglio, Maxime Hamadi, Vincent Drouet, Jean-Christophe Giret, et al. Industrialization of MDO Methods, Software and Processes, from MBSE to High Fidelity Multi-Components Wing, Pylon and Nacelle Optimisation. In *AIAA AVIATION FORUM AND ASCEND 2024*, page 4404, 2024.
- [25] Christopher A Lupp and Alexander Xu. Creating a Universal Communication Standard to Enable Heterogeneous Multidisciplinary Design Optimization. In *AIAA SCITECH 2024 Forum*, page 1799, 2024.