



HAL
open science

Modélisation magnéto-mécanique d'un nano commutateur. Optimisation sous contraintes de fiabilité par dérivation automatique des programmes en Java

Phuong Pham Quang Pham-Quang

► **To cite this version:**

Phuong Pham Quang Pham-Quang. Modélisation magnéto-mécanique d'un nano commutateur. Optimisation sous contraintes de fiabilité par dérivation automatique des programmes en Java. Autre. Université de Grenoble, 2011. Français. NNT : 2011GRENT060 . tel-00721947

HAL Id: tel-00721947

<https://theses.hal.science/tel-00721947>

Submitted on 31 Jul 2012

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.

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Génie Electrique**

Arrêté ministériel : 7 août 2006

Présentée par

Phuong PHAM-QUANG

Thèse dirigée par **Jean-Louis COULOMB** et
Co-encadrée par **Benoît DELINCHANT**

préparée au sein du **Laboratoire de Génie Electrique de Grenoble**
dans l'**École Doctorale Electronique, Electrotechnique,**
Automatique & Traitement du signal

**Modélisation magnéto-mécanique
d'un nano commutateur.
Optimisation sous contraintes de
fiabilité par dérivation automatique
des programmes en Java**

Thèse soutenue publiquement le **11 Octobre 2011**,
devant le jury composé de :

M. Laurent KRÄHENBÜHL

Directeur de recherche au CNRS, Président

M. Pascal BROCHET

Professeur à l'École Centrale de Lille, Rapporteur

M. Claude MARCHAND

Professeur à l'Université Paris-Sud 11, Rapporteur

M. Jean-Louis COULOMB

Professeur à Grenoble INP, Membre

M. Laurent HASCOËT

Directeur de recherche à l'INRIA, Membre

M. Laurent DURAFFOURG

Chercheur au CEA-MINATEC-LETI, Membre

M. Vincent LECONTE

Directeur technique de la Société CEDRAT S.A, Membre

M. Benoît DELINCHANT

Maître de conférences à l'UJF, Membre



Remerciements

Les travaux présentés dans ce manuscrit ont été effectués au Laboratoire de Génie Electrique de Grenoble (G2Elab) et à CEDRAT S.A dans le cadre d'une convention Industrielle de Formation par la Recherche (CIFRE).

J'adresse mes sentiments respectueux et reconnaissants à tous les membres du jury pour l'intérêt qu'ils ont porté à ce travail et l'honneur qu'ils m'ont fait de juger cette thèse :

Monsieur Laurent KRÄHENBÜHL, Directeur de recherche au CNRS, Responsable de l'équipe *Modélisation multi-échelle pour la conception* au Laboratoire Ampère à Lyon.

Monsieur Claude MARCHAND, Professeur à l'Université Paris-Sud 11, Responsable du département *Modélisation et Contrôle de Systèmes Électromagnétiques* au Laboratoire de Génie Electrique de Paris (LGEP).

Monsieur Pascal BROCHET, Professeur à l'Ecole Centrale de Lille, Responsable de l'équipe *Optimisation* au Laboratoire d'Electrotechnique et d'Electronique de Puissance de Lille (L2EP).

Monsieur Jean-Louis COULOMB, Professeur à l'Institut Polytechnique de Grenoble, Responsable de l'*Equipe Recherche Technologique (ERT) Champs Magnétiques Faibles (CMF)* au G2Elab.

Monsieur Laurent HASCOËT, Directeur de recherche à l'Institut National de Recherche en Informatique et en Automatique (INRIA), Responsable de l'équipe de recherche TROPICS.

Monsieur Laurent DURAFFOURG, Chercheur au CEA-MINATEC-LETI Grenoble.

Monsieur Vincent LECONTE, Directeur technique de la Société CEDRAT S.A.

Monsieur Benoît DELINCHANT, Maître de conférences à l'Université Joseph Fourier.

Je tiens à remercier vivement Monsieur Jean-Louis COULOMB, mon directeur de thèse qui, tout au long de ces trois ans, a su me laisser suffisamment de liberté tout en étant toujours présent lorsque cela était nécessaire. J'ai eu l'honneur et la chance d'avoir travaillé avec lui. Et j'espère de travailler encore avec lui dans les années à venir.

J'aimerais remercier particulièrement à Monsieur Benoît DELINCHANT, co-encadrant de cette thèse pour l'aide précieuse qu'il m'a apporté tout au long de ces travaux, pour m'avoir écouté et encouragé tout au long de ces travaux. J'ai eu l'occasion d'avoir travaillé avec un super débogueur.

J'aimerais remercier aussi à Monsieur Vincent LECONTE et la direction la société CEDRAT pour m'avoir laissé définir la thèse au G2Elab après des départs de mes chefs industriels que j'aimerais remercier également à eux : Guillaume LACOMBE, Bertrand DU PELOUX et Yann LE FLOCH.

Mes très vifs remerciements vont aussi à : Monsieur James ROUDET, Directeur de G2Elab, pour m'avoir accueilli au sein du laboratoire et à Monsieur Bruno RIBARD, Président Directeur Général de CEDRAT S.A pour m'avoir embauché dans sa société.

Je souhaite également remercier à Monsieur Gérard MEUNIER, à Monsieur Patrice LABIE, à Monsieur Nicolas GALOPIN, à Monsieur Christophe GUERIN, à Monsieur Jean-Michel GUICHON, à Monsieur Laurent GERBAUD, à Madame Laurence ESTRABAUT et à Monsieur Orphée CUGAT pour toutes les discussions, les aides et les conseils ils ont apportés.

Je remercie aussi à Mademoiselle Lise BILHAUT et à Monsieur Laurent DURAFFOURG pour nos coopérations dans le cadre du projet ANR MONACO, à Monsieur Emil-Ioan SLUSANSCHI pour son aide avec son outil ADiJaC.

Je remercie chaleureusement à mes amis Vietnamiens AEEE à Grenoble, à mes camarades de promotion K47 de l'Institut Polytechnique de Hanoi pour tous les agréables moments passés ensemble.

Je réserve les derniers mots pour ma famille : les grands soutiens de mes parents de mon grand frère m'ont encouragé de finir la thèse dans un délai impartie et de continuer de sacrifier pour la recherche.

*Ngày ta bảo vệ rất chi oai
Cũng gọi đóc tờ chẳng kém ai
Khổ luyện ba năm nên sự nghiệp
Rồi đây lại cốt tiếp ngày mai !*

Table des matières

Introduction générale	7
Contexte de la thèse.....	7
Contributions de la thèse	8
Organisation du manuscrit	8
I Chapitre I	11
Outils de conception de MEMS.....	11
I.1. Les MEMS et le nano commutateur magnétique	13
I.1.1. Actionnements dans les MEMS.....	13
I.1.2. Des MEMS aux NEMS.....	15
I.1.3. Le projet MONACO.....	16
I.1.4. Le nano commutateur magnétique.....	18
I.1.4.1. Principe de fonctionnement	18
I.1.4.2. Applications	18
I.1.4.3. Cahier de charge et enjeux de modélisation.....	19
I.2. Outils de modélisation de MEMS.....	20
I.2.1. Modélisation par éléments finis multi-physiques	20
I.2.2. Modélisation semi analytique	21
I.2.2.1. Environnements génériques.....	21
I.2.2.2. Environnements métiers CADES.....	22
I.2.2.3. MacMMems.....	23
I.3. Outils d'analyse	24
I.3.1. Analyse d'incertitude	24
I.3.2. Analyse de sensibilité	26
I.3.3. L'optimisation, une solution de conception	27
I.3.3.1. L'optimisation sous contraintes	27
I.3.3.2. L'optimisation sous contraintes de fiabilité	30
I.3.4. Dérivation des modèles de dimensionnement.....	31
I.3.4.1. Technique de calcul des gradients.....	31
I.3.4.2. Dérivation de modèle de dimensionnement dans CadesGenerator	33
I.4. Conclusions.....	34

II	Chapitre II	35
	Modélisation magnéto-mécanique du nano commutateur	35
	II.1. Modèle magnétique	37
	II.1.1. Modèle magnétique dans MacMMems.....	37
	II.1.1.1. Hypothèses	38
	II.1.1.2. Les conducteurs.....	39
	II.1.1.3. Les aimants	39
	II.1.2. Modèles magnétiques spécifiques (FM/AF)	43
	II.2. Modèle de force de Casimir	44
	II.3. Modèle mécanique	44
	II.3.1. Hypothèses.....	45
	II.3.2. Réduction de la poutre de 3D en 1D	46
	II.3.2.1. Cas d'une poutre parallélépipédique	46
	II.3.2.2. Cas d'une poutre constituée de plusieurs parallélépipèdes de même longueur	47
	II.3.2.3. Cas d'une poutre constituée de plusieurs parallélépipèdes	49
	II.3.3. Résolution de l'équation d'Euler - Bernoulli.....	49
	II.3.4. Analyse du contact mécanique.....	50
	II.3.5. Validation par éléments finis.....	55
	II.4. Couplage magnéto-mécanique	57
	II.4.1. Notions de couplage multi-physiques	57
	II.4.2. Couplage magnéto-mécanique.....	57
	II.5. Introduction du composant déformable dans MacMMems	59
	II.5.1. Interface Homme / Machine (IHM).....	59
	II.5.2. Visualisation de la déformation	62
	II.6. Conclusions	63
III	Chapitre III	65
	JAP : Java Jacobian Automatic Programming	65
	III.1. Motivations pour JAP	67
	III.1.1. Introduction	67
	III.1.2. Outils existants de dérivation automatique en Java.....	67
	III.1.3. Motivation pour JAP.....	69
	III.2. Principes et techniques de Dérivation Automatique	69

III.2.1. Les modes de Dérivation Automatique	70
III.2.1.1. <i>Le mode direct</i>	71
III.2.1.2. <i>Le mode inverse</i>	72
III.2.2. Les stratégies d'implémentation de la DA	73
III.2.2.1. <i>Surcharge d'opérateurs</i>	73
III.2.2.2. <i>Génération de code</i>	74
III.3.Principe, architecture et limitation de JAP	76
III.3.1. Principe de DA de JAP en mode direct.....	76
III.3.1.1. <i>Objet derivative ∇u de JAP : Jdouble1</i>	76
III.3.1.2. <i>Propagation de j-méthodes</i>	77
III.3.1.3. <i>Exploitation du Jacobien et calcul de dérivées partielles</i>	79
III.3.1.4. <i>Gestion de sous fonctions externes</i>	80
III.3.2. Architecture de JAP	82
III.3.3. Limitations de JAP	84
III.4.Performances de JAP	85
III.5.Usages de JAP.....	88
III.5.1. Mise en oeuvre de JAP dans CADES.....	88
III.5.1.1. <i>Description d'un modèle de dimensionnement dans CADES</i>	88
III.5.1.2. <i>JAP_ExternalFunctionGenerator : Générateur de fonctions externes algorithmiques</i> 91	
III.5.1.3. <i>JAPGenerator : Générateur de modèle de dimensionnement sml</i>	92
III.5.2. Exemples d'utilisation de JAP dans CADES.....	94
III.5.2.1. <i>Validation et performances de JAP dans CADES</i>	94
III.5.2.2. <i>Modélisation et optimisation du système de suspension magnétique..</i>	96
III.5.2.3. <i>Utilité d'un code 100% Java</i>	99
III.5.2.4. <i>Optimisation du nano commutateur magnétique</i>	101
III.5.3. Utilisation de JAP pour la simulation dynamique	102
III.6.Conclusions.....	104
IV Chapitre IV	107
Optimisation sous contraintes de fiabilité	107
IV.1.Analyse de sensibilité et propagation d'incertitude.....	109
IV.1.1. Méthodes d'analyse	109
IV.1.1.1. <i>Approximation linéaire</i>	109
IV.1.1.2. <i>Méthode de Monte-Carlo</i>	112
IV.1.2. Implémentation dans CADES	112
IV.1.3. Application : analyse du nano commutateur magnétique	113
IV.2.Optimisation sous contraintes du nano commutateur magnétique	116

IV.2.1. Cahier des charges	116
IV.2.2. Stratégies d'optimisation.....	117
IV.2.2.1. Optimisation directe par algorithme stochastique	117
IV.2.2.2. Optimisation indirecte par surface de réponse adaptative	118
IV.2.2.3. Optimisation par un algorithme déterministe exploitant le Jacobien	119
IV.2.3. Résultats d'optimisation.....	120
IV.2.4. Sensibilité de la solution optimale	122
IV.3. Optimisation sous contraintes de fiabilité	123
IV.3.1. Contrôle de qualité : Six sigma (6σ).....	123
IV.3.2. Méthode découplée de résolution du problème d'optimisation sous contraintes de fiabilité.....	125
IV.3.3. CadesDeviationGenerator.....	127
IV.3.3.1. Principe de CadesDeviationGenerator.....	127
IV.3.3.2. Composition du Jacobien dans CadesDeviationGenerator.....	129
IV.3.4. Application pour le nano commutateur magnétique	131
IV.3.4.1. Résultats d'optimisation	131
IV.3.4.2. Vérification des résultats par Monte-Carlo.....	132
IV.4. Conclusions.....	134
Conclusion générale et perspectives.....	135
Références bibliographiques	137
Publications	147
Annexes	149
A. Jap Exceptions	149
B. Services de calcul de sensibilité dans CadesCalculator.....	151
C. Géométries et paramétrage du nano commutateur magnétique	154
D. Résultats de comparaison entre JAPGenerator et AdolCGenerator sur les cas test de CADES	156

Introduction générale

Contexte de la thèse

Cette thèse s'inscrit dans le cadre du projet de l'Agence National de la Recherche -ANR intitulé MONACO et s'intéressant à la « Modélisation et Exploitation des phénomènes physiques nanométriques intervenant dans les NEMS ». Ce projet vise à appréhender et à modéliser le rôle de certains effets submicroniques sur les nano structures mécaniques et à les quantifier sur des structures tests génériques afin de concevoir de nouveaux outils de simulation et de conception adaptés. Ce projet regroupe plusieurs partenaires dont le CEA/LETI et G2Elab/CEDRAT. Le rôle de G2Elab/CEDRAT a été de développer un outil de couplage magnéto-mécanique (basé sur le logiciel CADES/MacMMems conçu au G2Elab) adapté à la modélisation et l'optimisation des NEMS magnétiques. La démarche a été appliquée à un nano commutateur magnétique [BIL-08], choisi comme cas test du projet, et dont la conception a été réalisée par notre partenaire le CEA-LETI.

CADES¹ est une suite logicielle dédiée au dimensionnement de systèmes développé au G2Elab [DEL-07]. Ses applications sont potentiellement multi physiques, puisque les modèles sont formulés sous forme d'équations analytiques (générés par des modules métiers). MacMMems² [RAK-07] est un des modules métiers de CADES dédié aux MEMS magnétiques. Pour la modélisation semi analytique et l'optimisation du nano commutateur magnétique, CADES/MacMMems présente certaines limites qu'il faudra lever :

✚ MacMMems : il permet de calculer semi-analytiquement des forces, des couples, des champs magnétiques sur des objets magnétiques non déformables comme des conducteurs ou des aimants. Pour réaliser la modélisation multi physiques des composants déformables dans MacMMems, il lui manque les modèles mécaniques et de couplage magnéto-mécanique.

✚ CADES : il permet de traiter des modèles formulés sous forme d'équations analytiques et d'algorithmes décrits en langage C (branches

¹ CADES : *Component Architecture for the Design of Engineering Systems*

² MacMMems : *Macro Modeller for Magnetic MEMS*

conditionnelles if...then...else, les boucles répétitives while...do, for, etc.,). Des solutions sont disponibles présentant certains inconvénients que nous avons cherché à éliminer en développant dans cette thèse une solution 100% Java (langage cible privilégié de CADES) permettant de simplifier la chaîne de dérivation automatique et d'étude de sensibilité dans le but de réaliser l'optimisation sous contraintes de fiabilité.

Contributions de la thèse

Dans ce contexte, cette thèse contribue à la modélisation et à la conception de système magnéto-mécanique selon trois axes :

✚ Axe modélisation : Développement et introduction du composant déformable de couplage magnéto-mécanique dans MacMMems, dédié au nano commutateur magnétique. Pour cela, un modèle semi analytique de calcul de déformation intégrant une analyse de contact a été développé puis validé par simulation numérique.

✚ Axe dérivation automatique de code : Développement d'un outil de dérivation automatique de code en Java (JAP- Java Jacobian Automatic Programming). Comme JAP est un outil générique, ses applications sont multiples et non limitées à CADES. JAP a d'ailleurs été exploité par d'autres personnes pour l'optimisation de systèmes ainsi que la réorientation d'équations différentielles pour la simulation dynamique.

✚ Axe optimisation : Mise en œuvre du modèle et des outils développés pour faire l'étude de sensibilité/incertitude et l'optimisation sous contraintes de fiabilité appliqué au nano commutateur magnétique comme un exemple de test.

Organisation du manuscrit

Ce document est structuré en quatre chapitres.

Le premier chapitre expose le cadre dans lequel s'inscrit ce travail de thèse. Les outils existants de conception de MEMS y sont présentés pour positionner les enjeux de la thèse.

Le deuxième chapitre traite de la modélisation semi analytique du couplage magnéto-mécanique dédié au nano commutateur magnétique. Le modèle magnétique existant dans MacMMems y est d'abord présenté brièvement. Ensuite, nous présentons le modèle mécanique développé calculant la déformation de la poutre avec analyse du

contact mécanique. La validation de ce nouveau modèle par une méthode éléments finis y est aussi présentée. Nous finissons enfin par le couplage magnéto-mécanique et son introduction dans le logiciel MacMMems.

Le troisième chapitre est consacré à l'outil de dérivation automatique de code. Après avoir présenté le principe et l'architecture de l'outil JAP développé, nous présentons ses applications dans des outils de dimensionnement du G2Elab.

Le quatrième chapitre concerne les méthodes et outils génériques d'analyse de sensibilité/incertitude et d'optimisation sous contraintes de fiabilité, que nous avons appliqués au nano commutateur magnétique.

Le manuscrit se termine par une conclusion générale donnant les perspectives de ces travaux de thèse.

Chapitre I

Outils de conception de MEMS

SOMMAIRE

Chapitre I

Outils de conception de MEMS.....	11
I.1. Les MEMS et le nano commutateur magnétique	13
I.1.1. Actionnements dans les MEMS.....	13
I.1.2. Des MEMS aux NEMS.....	15
I.1.3. Le projet MONACO.....	16
I.1.4. Le nano commutateur magnétique.....	18
I.1.4.1. Principe de fonctionnement	18
I.1.4.2. Applications	18
I.1.4.3. Cahier de charge et enjeux de modélisation.....	19
I.2. Outils de modélisation de MEMS.....	20
I.2.1. Modélisation par éléments finis multi-physiques	20
I.2.2. Modélisation semi analytique	21
I.2.2.1. Environnements génériques.....	21
I.2.2.2. Environnements métiers CADES.....	22
I.2.2.3. MacMMems.....	23
I.3. Outils d'analyse	24
I.3.1. Analyse d'incertitude	24
I.3.2. Analyse de sensibilité	26
I.3.3. L'optimisation, une solution de conception	27
I.3.3.1. L'optimisation sous contraintes	27
I.3.3.2. L'optimisation sous contraintes de fiabilité	30
I.3.4. Dérivation des modèles de dimensionnement.....	31
I.3.4.1. Technique de calcul des gradients.....	31
I.3.4.2. Dérivation de modèle de dimensionnement dans CadesGenerator	33
I.4. Conclusions.....	34

Résumé du chapitre

Ce chapitre vise à préciser le cadre dans lequel s'inscrit ce travail. Après une brève présentation des principes d'actionnement dans les MEMS et du nano commutateur magnétique, les outils existants de modélisation des MEMS sont introduits pour mettre en évidence la nécessité de développer un outil spécifique dédié au nano commutateur magnétique. Nous introduisons ensuite les outils d'analyse comme l'analyse d'incertitude, l'analyse de sensibilité, l'optimisation sous contraintes, l'optimisation sous contraintes de fiabilité. Nous détaillons enfin les techniques pour évaluer le gradient, qui est utile pour tous ces types d'analyse.

I.1. Les MEMS et le nano commutateur magnétique

MEMS est l'acronyme anglais de Micro-Electro-Mechanical Systems. Les MEMS ont pour caractéristique d'être de taille submillimétrique et d'être fabriqués de manière collective par des procédés de microélectronique : lithographie, dépôt, gravure. Leur spécificité est que le système présente une partie dite libérée, i.e. capable de bouger sous l'action des forces et des couples extérieurs (thermique, électrique, magnétique, etc.). Le premier MEMS date des années 60 [NAT-67] avec la fabrication d'un transistor à grille métallique résonante, mais jusqu'au début des années 80, les propriétés mécaniques du silicium n'ont pas été vraiment étudiées [PET-82]. L'immense succès du silicium pour la réalisation de microsystème provient du fait qu'il est le matériau de base pour la co-intégration de structures mécaniques et de fonctions électriques.

On peut diviser les MEMS en trois catégories : les actionneurs, les capteurs et les générateurs. Pour les actionneurs, on peut citer des relais (pour l'électronique de puissance ou pour des applications optiques), des moteurs ou encore des valves etc. Pour les capteurs, on peut citer des accéléromètres (utilisés dans les systèmes d'airbag, dans les manettes de jeux, ou encore dans la navigation), des capteurs de pression, des capteurs thermiques, chimiques, etc. Grâce à leur petite taille et à leur méthode de fabrication, les MEMS permettent la réduction des coûts, des masses, de l'encombrement ou de l'énergie utilisée. Ces avantages sont primordiaux dans des systèmes embarqués ou autonomes tels que les satellites, les téléphones portables, etc. De plus, l'extrême réduction de taille permet d'exploiter de propriétés amplifiées (ie. augmentation des effets surfaciques par rapport aux effets volumiques) qui ne se manifestent qu'aux très petites échelles.

I.1.1. Actionnements dans les MEMS

Nous présentons ici brièvement les différents moyens d'actionnement dans des microsystèmes. Pour plus de détails sur les principes abordés, nous recommandons au lecteur l'ouvrage de référence [GIA-08].

✚ Actionnement électrostatique : ce type d'actionnement exploite le principe d'un condensateur variable. La force électrostatique F_{ES} entre deux plaques en regard de même surface S et séparées d'une distance g s'exprime par :

$$F_{ES} = \frac{\epsilon_0 S V^2}{2g^2} \quad \text{Eq. I-1}$$

où ϵ_0 est la perméabilité du vide et V la tension appliquée aux bornes des deux plaques.

Les énergies consommées sont assez faibles car uniquement dues au courant de fuite ou au courant nécessaire pour charger les capacités. L'inconvénient de ce type d'actionnement sont, d'une part, que la bistabilité est difficile à mettre en œuvre, et, d'autre part, qu'il requiert des champs électrostatiques élevés (et donc des tensions d'alimentation élevées). A cause de ces tensions élevées, le système est susceptible d'être dégradé par le claquage (arc électrique) dans les diélectriques. Malgré ces inconvénients, les MEMS électrostatiques sont très répandus sur le marché car ils utilisent des matériaux et des techniques de fabrication très simples et très courants en microélectronique.

✚ Actionnement thermique : ce type d'actionnement utilise la déformation de la matière sous l'effet de la chaleur. On associe plusieurs matériaux avec différents coefficients d'expansion thermique. Un effet bilame résulte des différences déformations qu'ils subissent. L'inconvénient de ce type d'actionnement est sa consommation importante en énergie et la difficulté à mettre en œuvre la bistabilité.

✚ Actionnement piézoélectrique : Les matériaux piézoélectriques (e.g. quartz, AlN, Pb(Zr,Ti)O₃) sont des matériaux qui se déforment lorsqu'on leur applique une tension. Les tensions appliquées sont faibles, mais les déplacements sont également très faibles et doivent être amplifiés par des designs appropriés, dont le plus connu dans les MEMS est le principe de bilame. On peut citer les applications des actionneurs piézoélectriques comme des micros pompes, des têtes d'imprimante, etc.

✚ Actionnement magnétostrictif : Les matériaux magnétostrictifs sont des matériaux qui se déforment en présence d'un champ magnétique. Les matériaux couramment utilisés sont des alliages de terres rares et de métaux de transition tel le Terfenol-D (TbDyFe). Le principe pour créer une déformation est le même que pour les matériaux piézoélectriques, à la différence que la source est un champ magnétique et non plus un champ électrique. Le principe du bilame est aussi couramment utilisé. Il permet de réaliser des actionneurs comme des micro pompes ou des capteurs de champs magnétiques.

✚ Actionnement magnétique : Ces moyens d'actionnement sont couramment utilisés dans le monde macroscopique et de plus en plus dans les MEMS.

La bistabilité peut s’obtenir assez facilement parce que les aimants produisent des forces permanentes, et la direction des forces de Laplace peut être inversée par l’inversion du courant. Dans le chapitre 2 de [REY-02], l’auteur démontre les lois de réduction d’échelle (résumées dans le Tableau I-1) des interactions entre des aimants permanents, des bobines, des matériaux ferromagnétiques et des courants induits par variation de champ magnétique. Ces lois de réduction d’échelle montre l’efficacité des interactions à base d’aimants dans les MEMS.















réduction échelle $1/k$	aimant 	courant 	fer 	induction 
aimant 	$\times k$ 		$\times k$ 	$/k$ 
courant 		$/k$ 	$/k$ 	$/k^2$ 

Tableau I-1. Effet de la réduction d’échelle sur la force d’interaction

Si les familles d’actionnement électrostatique et piézoélectrique trouvent leur intérêt dans les domaines du petit déplacement et les dispositifs hautes fréquences, la famille d’actionnement magnétique est plus adaptée pour des applications demandant une force d’actionnement conséquente avec un plus grand déplacement et ne nécessitant pas d’alimentation haute tension [NIA-03]. Depuis, d’autres travaux sur l’intégration d’aimant par couches ferromagnétiques/antiferromagnétiques ont montré l’intérêt des interactions magnétiques à l’échelle sub-micrométrique [VIA-08].

I.1.2. Des MEMS aux NEMS

Les NEMS (Nano Electro-Mechanical Systems) sont des dispositifs analogues aux MEMS, mais à l’échelle du nanomètre. On parle de NEMS pour des structures mécaniques miniatures, dont au moins une des dimensions est de taille nanométrique. L’intérêt pour les NEMS date de la fin des années 90 [CLE-99], mais reste un domaine de recherche jeune et en pleine expansion.

Les applications de MEMS ne sont possibles que grâce à la réduction des coûts. Cette baisse des coûts peut passer par une amélioration des procédés, augmentant ainsi le rendement de production. Sachant que le packaging est estimé à 80% du prix des MEMS [BIL-09], un travail sur l’intégration des MEMS dans des boîtiers à bas coûts

peut participer à la diminution des coûts. Les NEMS offrent un autre avantage car leurs surfaces étant 100 à 500 fois inférieures à celles des MEMS, plus de capteurs peuvent être fabriqués sur un même substrat.

En complément d'une réduction des coûts, la réduction des tailles a deux effets : tout d'abord l'augmentation de la fréquence de résonance pour les capteurs dynamiques ce qui permet de fabriquer des capteurs ultrasensibles ; ensuite la consommation électrique réduite ce qui est utile pour toutes les technologies nomades comme la téléphonie mobile ou les réseaux de capteurs autonomes.

En raison de leur taille, la physique des NEMS est différente de la physique « à l'échelle humaine ». Le passage des MEMS aux NEMS soulève plusieurs difficultés comme cela est montré dans [BIL-09] :

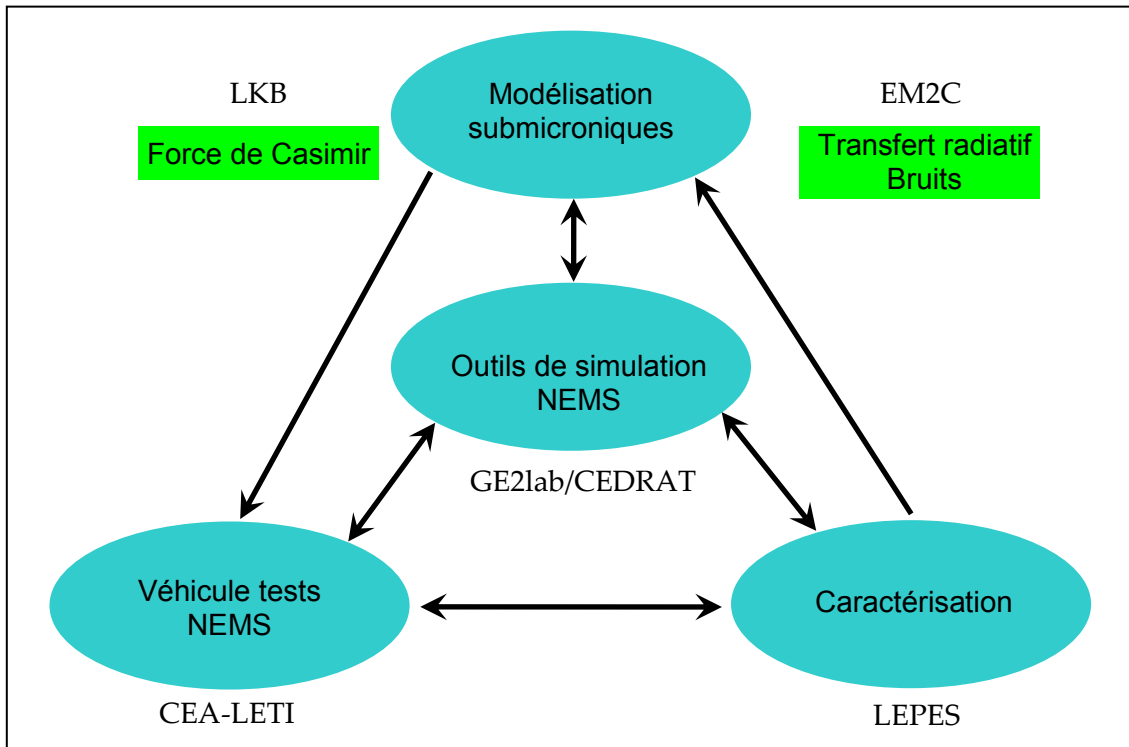
✚ Au niveau scientifique, le passage des MEMS aux NEMS ne peut pas se réduire à un simple changement d'échelle. Comme le rapport de la surface sur le volume augmente, les effets dus à cette surface deviennent prépondérants. Des forces négligeables dans le monde macroscopique doivent être prises en considération dans la conception des NEMS) : forces de proximité comme la force de Casimir ou celles de van der Waals, force de capillarité, effets de charges. Comme le rapport signal à bruit décroît avec la taille, la problématique de la détection est essentielle. Les mouvements à détecter sont très faibles, alors que les sources de bruit à prendre en compte sont plus nombreuses (bruit thermomécanique, bruit en $1/f$, bruit de Johnson, bruits liés à la surface comme les phénomènes d'adsorption et de désorption des molécules de gaz, etc.) [BAR-08].

✚ Au niveau technologique, la reproductibilité des systèmes est essentielle pour la production de masse. Les marges de fonctionnement du dispositif doivent donc être d'autant plus importantes. La lithographie induit également une grande dispersion. Cette dispersion est par la suite aggravée par les inhomogénéités dues à la gravure [BIL-09]. Comme la surface joue un rôle prépondérant, les procédés qui jusque là étaient suffisants pour les MEMS doivent être optimisés. Des méthodes de validation dimensionnelle et d'observation non destructive nécessitent également d'être mises en place.

I.1.3. Le projet MONACO

Comme nous avons mentionné dans l'introduction générale, le point de départ de cette thèse est le projet MONACO. Ce projet vise à appréhender le rôle de certains effets submicroniques sur les nano structures mécaniques et à les quantifier sur des

structures tests génériques pour une modélisation et une conception pertinente des futurs nano systèmes. Les partenaires du projet sont décrits dans la figure suivante.



LKB : Laboratoire Kastler Brossel

EM2C : Laboratoire d'Energétique Moléculaire, Macroscopique, Combustion

LEPES : Laboratoire d'Etudes des Propriétés Electroniques des Solides (Grenoble)

CEA-LETI : Commissariat à l' Energie Atomique et aux Energies Alternatives – Laboratoire d' Electronique et de Technologie de l'Information

Figure I-1. Partenaires du projet MONACO

Le premier objectif du projet est de développer et de valider des modèles et des méthodes expérimentales originales permettant d'étudier ces phénomènes submicroniques dans le contexte des NEMS.

Le second objectif du projet est de développer et d'implémenter un cœur de programme de conception de nano systèmes pour le développement des futurs composants mécaniques. A partir du cas d'étude d'un nano commutateur magnétique fournit par le CEA-LETI, nous (G2Elab/CEDRAT) devons développer un outil semi analytique adapté à la modélisation et l'optimisation des NEMS magnétiques sur la base de CADES/MacMMems.

I.1.4. Le nano commutateur magnétique

I.1.4.1. Principe de fonctionnement

Le principe de fonctionnement du nano commutateur magnétique est illustré dans la Figure I-2 et est détaillé dans [BIL-08].

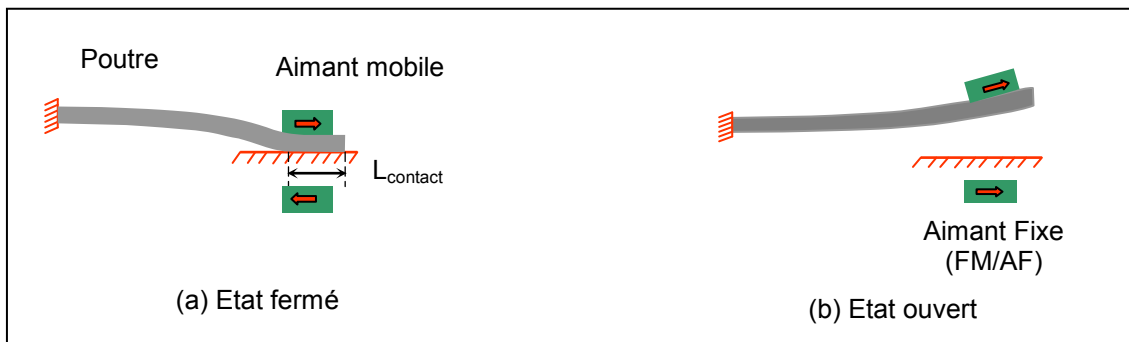


Figure I-2. Principe de fonctionnement du nano commutateur magnétique

Un aimant permanent est placé sur une poutre à simple encastrement. Cette poutre est déformée sous l'interaction entre cet aimant mobile et un empilement spécifique ferromagnétique/antiferromagnétique (FM/AF). La commutation est réalisée par l'utilisation du conducteur pour chauffer et inverser la magnétisation de la couche FM/AF. Cette procédure de commutation sera détaillée dans la section II.1.2 du chapitre II.

I.1.4.2. Applications

Les nano commutateurs magnétiques sont en cours d'étude, ils n'existent pas encore en tant que produits industriels, mais beaucoup de domaines d'application ont été identifiés [BIL-09] comme par exemple :

✚ Les interrupteurs d'alimentation pour circuit basse consommation (ou power gating). Ces interrupteurs permettent de couper des tensions et des courants d'alimentation afin de supprimer les courants de fuite. Cette technique, utile notamment pour les applications nomades, exige que les interrupteurs eux-mêmes présentent un faible courant de fuite, ce qui devrait être le cas des nano commutateurs magnétiques par rapport aux interrupteurs MOS¹ classiques.

✚ Les convertisseurs DC/DC : Les nano commutateurs magnétiques sont des bons candidats pour remplacer les transistors MOS dans les convertisseurs DC/DC qui

¹ MOS : Metal-Oxyde-Semiconductor

permettent de hacher l'énergie disponible d'une source donnée dans une forme utilisable par sa charge.

✚ Dans des circuits à capacités commutées : ils représentent une catégorie importante des circuits analogiques de traitement du signal qui sont actuellement réalisés avec des CMOS¹. Mais certains phénomènes dus à ces CMOS dégradent les performances, comme la réduction de la bande passante et des phénomènes de distorsion, ce qui pourraient être évité par l'utilisation des nano commutateurs magnétiques.

✚ Les mémoires nano mécaniques dans lesquels un bit d'information est représenté par l'état mécanique d'un élément.

✚ Les commutateurs RF, les actionneurs pour membranes déformables, etc.

I.1.4.3. Cahier de charge et enjeux de modélisation

Les applications potentielles du nano commutateur, qui ont été présentées dans la section précédente, nous ont permis de dresser un cahier des charges générique que nous détaillons au Tableau I-2 [BIL-09], [COL-10].

Résistance de contact à l'état ON	$0,1 < R_{ON} < 100 \Omega$
Consommation de maintien	$\ll \mu A$
Capacité parasite	$< pF$
Consommation de commutation	?
Temps de commutation	10 – 100 μs (power gating) 10 – 100 ns (DC/DC)
Dimension totale	$< 100 \times 100 \mu m^2$

Tableau I-2. Cahier de charge du nano commutateur

Le bon fonctionnement du nano commutateur est dominé par la qualité de son contact. D'une manière générale, cette qualité de contact est gouvernée par un certain nombre de facteurs [PEN-09] :

- ✓ Force de contact
- ✓ Surface de contact
- ✓ Matériaux de contact (couche(s) mince(s) et substrat)
- ✓ Intensité de courant traversant le contact
- ✓ Environnement de fonctionnement du nano commutateur

¹ CMOS : Complementary Metal-Oxyde-Semiconductor

Pour modéliser le nano commutateur magnétique afin d'évaluer la qualité de contact, l'outil utilisé devrait réaliser un couplage magnéto-mécanique avec l'analyse de contact. De plus cet outil devrait être capable de tenir compte des forces externes présentes en taille nanométrique, comme la force de Casimir ou celles de van der Waals etc.

D'autres aspects du cahier des charges, ne sont pas des objectifs, mais plutôt des perspectives de cette thèse comme la modélisation dynamique pour connaître le temps de réponse, etc.

I.2. Outils de modélisation de MEMS

I.2.1. Modélisation par éléments finis multi-physiques

L'analyse par la méthode des éléments finis ou FEM est un outil mathématique qui a été développé à l'origine pour le calcul de structures mécaniques complexes et pour en évaluer la résistance à la rupture. Ces outils sont aujourd'hui très utilisés dans le domaine de la mécanique, du génie civil, du génie électrique, etc. Il est possible grâce à ces méthodes numériques d'effectuer des analyses de structures en mode statique ou dynamique dans les domaines linéaires et non linéaires, mais également des analyses thermiques, de mécanique des fluides et d'électromagnétisme.

La nature multidisciplinaire des MEMS fait que ces outils d'analyse sont très utilisés pour la conception des micro-systèmes. En effet, le comportement mécanique des dispositifs MEMS est une conséquence directe des phénomènes physiques mis en jeu (cela dépend du type d'actionnement cité dans la section I.1.1), comme l'électrostatique, la thermique, etc. Ce couplage du comportement mécanique avec un autre effet physique, combiné avec des considérations liées à leurs petites dimensions, pose des défis importants à l'analyse des MEMS. Ces défis peuvent être relevés par les logiciels FEM capables d'effectuer l'analyse multi physique. Avec ces logiciels à leur disposition, les ingénieurs peuvent utiliser les résultats de leurs simulations pour concevoir des dispositifs MEMS.

Des logiciels correspondants sont disponibles sur le marché et sont utilisés par les communautés de concepteurs de micro-systèmes. On peut citer des solutions commerciales comme IntelliSuite® de IntelliSense Software Corp [INT], CoventorWare® de Coventor Inc [COV] et MEMS Pro® de SoftMEMS LLC [MEM] qui sont les outils de CAO dédiés particulièrement aux MEMS.

Avec l'augmentation du marché des MEMS, des fournisseurs de logiciels commerciaux pour la simulation multi-physiques générique ont fourni des outils intégrés pour des applications MEMS comme Ansys® de ANSYS Inc [ANS], Comsol® de COMSOL, Inc [COM], Oofelie® de Open Engineering S.A [OOF].

Malgré cette panoplie d'outils, aucun d'entre eux ne permet, à notre connaissance, un couplage magnéto-mécanique avec l'analyse de contact. Pour modéliser le nano commutateur, il faut tenir compte des déformations et du contact au couplage magnéto-mécanique et il faut coupler au modèle de force magnétique un modèle complémentaire de forces (forces d'adhésion). Compte tenu de la difficulté de cette modélisation et des particularités du dispositif à modéliser, nous nous sommes orientés vers une modélisation semi analytique. Nous verrons que cette modélisation semi analytique est importante pour réaliser le dimensionnement sous contraintes de fiabilité développé au chapitre IV.

I.2.2. Modélisation semi analytique

I.2.2.1. Environnements génériques

Plusieurs environnements génériques sont de bons candidats pour modéliser des MEMS comme :

✚ Des environnements de calcul matriciel comme Matlab et/ou ses alternatives Scilab [SCI] et Octave [OCT], qui sont destinés essentiellement à réaliser des traitements numériques, la visualisation et l'analyse des données, et l'édition de nouveaux programmes utilisateurs. Avec l'ensemble des produits associés et complémentaires à Matlab, plusieurs possibilités et services permettent de modéliser des MEMS.

✚ Des environnements de calcul algébrique formel comme Mathematica [WOL] ou Maple [KOF], qui sont destinés à la modélisation analytique grâce à leurs capacités de traitement symbolique d'équations. On dispose de modules spécifiques comme Radia [RAD] pour le calcul de champ magnétique.

A ces environnements génériques, nous avons préféré le framework CADES pour les raisons que nous allons présenter maintenant.

I.2.2.2. Environnements métiers CADES

L'environnement standard de CADES est une suite logicielle dédiée au dimensionnement de systèmes, contrairement aux environnements génériques précédents. Son application est potentiellement multi-physiques, puisque tout modèle formulable sous forme d'équations analytiques et d'algorithmes peut être traité par l'environnement CADES.

CADES est fractionné en trois modules qui sont :

- ✓ Le générateur : *CadesGenerator*
- ✓ La calculette : *CadesCalculator*
- ✓ L'optimiseur : *CadesOptimizer*

Le point d'entrée dans l'environnement est le module *CadesGenerator*. L'utilisateur fournit un modèle (équations analytiques, fonctions externes (.jar¹) – en Java), des traitements lui sont automatiquement appliqués par le générateur afin de produire en sortie un modèle de type boîte noire : composant logiciel de calcul (ICAr²).

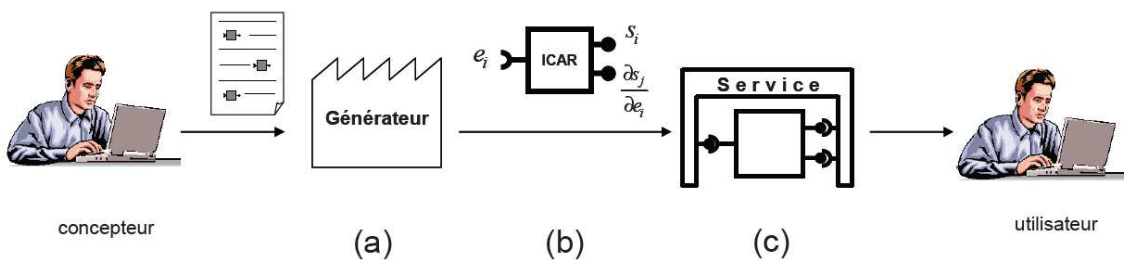


Figure I-3. A partir d'un modèle boîte grise (conçu par le concepteur),

Le générateur (a) crée le composant de calcul (ICAr) (b),

Ce dernier est exploité (par l'utilisateur) via le service(c)

Les services disponibles dans l'environnement standard sont *CadesCalculator* et *CadesOptimizer*.

Le *CadesCalculator* est une sorte de calculette utilisé pour étudier au coup par coup le comportement du modèle : grâce à lui, l'utilisateur peut dialoguer avec le composant de calcul, entrer des jeux de valeurs et observer directement leur influence en sortie. Elle permet aussi de tracer la caractéristique d'une grandeur de sortie en fonction d'une ou plusieurs grandeurs d'entrée.

¹ jar : Java Archive, type de fichier comme ZIP, utilisé pour distribuer un ensemble de classes Java

²ICAr : Interfaces for Component Architecture – Le nom de la norme du composant logiciel de CADES

Une fois que le concepteur s'est familiarisé avec son modèle (via les services proposés par la calculette), il peut passer à l'étape d'optimisation. Ce dernier est implémenté par le service d'optimisation que propose *CadesOptimizer* (Figure I-4). Il est également possible d'importer le modèle (composant de calcul ICAR) dans le logiciel d'optimisation *FGOT* (Figure I-5) [COU].

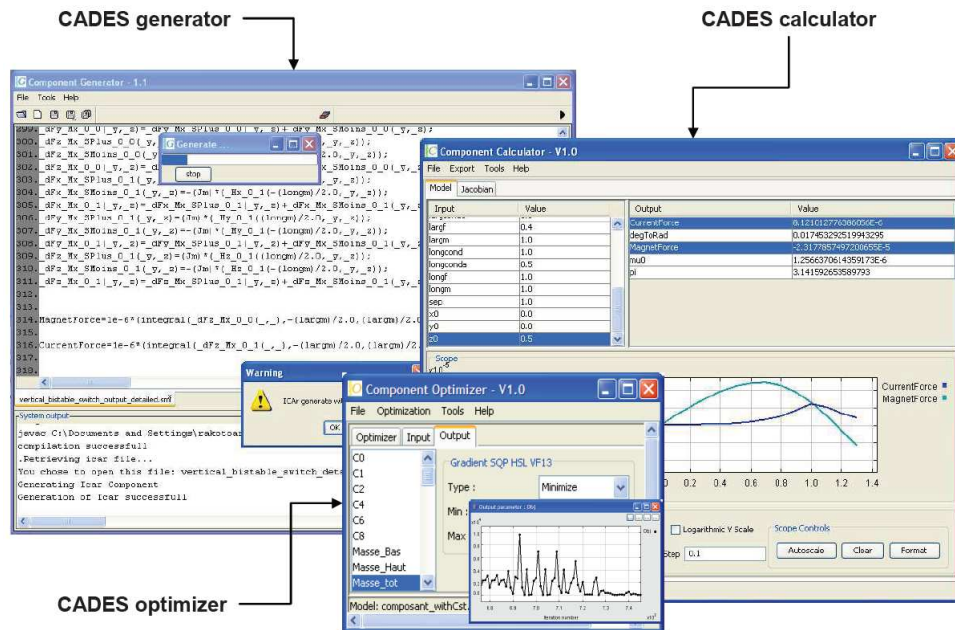


Figure I-4. IHM de framework CADES comprenant le générateur et les services

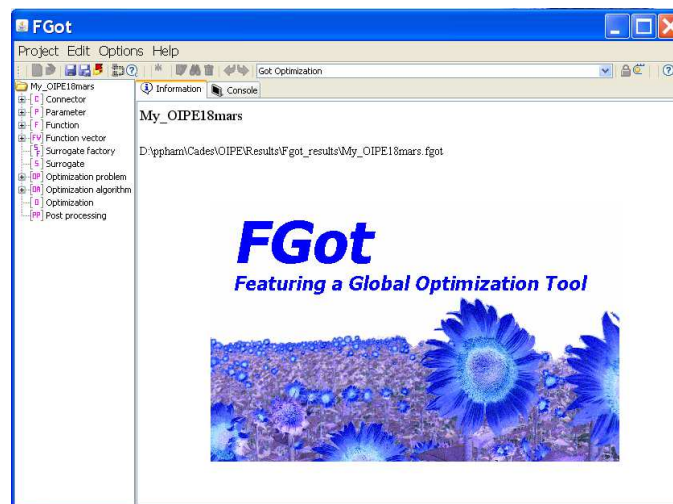


Figure I-5. IHM du logiciel d'optimisation *FGOT*

I.2.2.3. MacMMems

De plus, en comparaison avec les environnements génériques, CADES dispose de MacMMems qui est un générateur de modèle dédié au MEMS magnétiques. Il a été développé durant la thèse de H. L. Rakotoarison au G2Elab [RAK-07]. Il crée les

modèles analytiques et semi-analytiques relatifs à chaque composant magnétique élémentaire (aimant, conducteur, ferromagnétique, diamagnétique). MacMMems calcule les forces, les couples, le champ magnétique, l'induction magnétique sur chaque composant. De plus, il permet aux utilisateurs d'ajouter des équations supplémentaires.

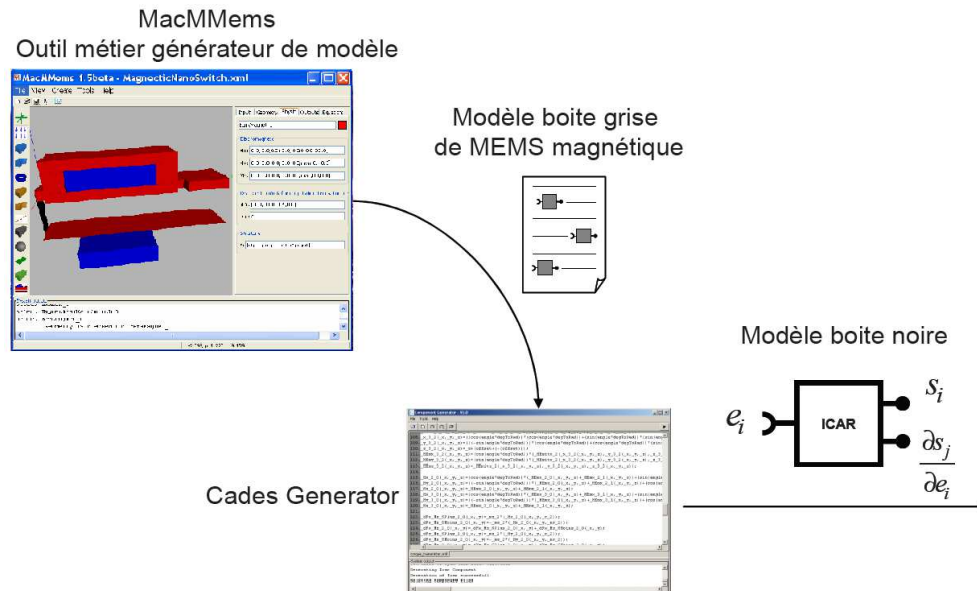


Figure I-6. Génération de modèles semi analytiques pour les MEMS magnétiques

I.3. Outils d'analyse

I.3.1. Analyse d'incertitude

L'analyse d'incertitudes consiste à étudier l'impact des facteurs incertains d'un modèle de dimensionnement sur ses sorties. Les incertitudes sont regroupées en deux grands types :

- ✚ Les incertitudes dites épistémiques, dues à la construction du modèle de dimensionnement. Par exemple, en modélisant on peut manquer de connaissance sur le phénomène ou disposer d'hypothèses de simplification qui ne rendent que partiellement compte de la réalité. Par ailleurs, on peut citer aussi le passage d'un modèle mathématique au modèle informatique correspondant, qui par exemple utilise un schéma numérique pour résoudre une équation différentielle de manière approximative. Pour réduire ce type d'incertitudes, on peut ajouter de l'information ou sacrifier la simplicité des modèles pour affiner la précision.

- ✚ Les incertitudes dites stochastiques (ou aléatoires), elles sont dues à la déviation naturelle inhérente à toute mesure expérimentale (erreurs de mesure et de

reproductibilité). Certaines de ces incertitudes peuvent être réduites (en améliorant la précision des instruments de mesure, en diminuant la tolérance de fabrication etc.).

C'est ce dernier type d'incertitude qui sera pris en compte dans notre étude. En particulier, nous chercherons à propager à travers le modèle des incertitudes telles que celles liées à la fabrication du dispositif que nous simulons.

Afin de propager ces incertitudes, différentes techniques sont envisageables [SAL-00]. Par exemple, dans un cadre déterministe, on voudra faire une étude des pires cas qui consiste à faire varier les facteurs incertains dans un intervalle donné, et d'en déduire un intervalle contenant les sorties. Cette méthode produit cependant des intervalles non-optimaux au sens probabiliste. Elle suppose implicitement que les facteurs peuvent varier uniformément sur leur intervalle de valeurs. C'est une hypothèse a priori très forte à vérifier.

D'une autre manière, dans un cadre probabiliste, on s'intéresse à la distribution de probabilité des sorties, à leurs premiers moments (moyenne et variance), à des intervalles de confiance associés à des risques ou encore à des probabilités de dépassement de seuil.

L'approche probabiliste se base sur un modèle de type :

$$Y = \eta(X) \tag{Eq. I-2}$$

où le vecteur X contient tous les facteurs incertains.

On associe alors une loi de probabilité au vecteur des facteurs X . Dans le cas où les facteurs sont indépendants, ceci revient à donner une loi de probabilité à chaque composante de X indépendamment. Par propagation à travers la fonction η , la sortie Y est elle aussi aléatoire et possède donc une loi de probabilité. A cette étape de l'analyse d'incertitude, il faut identifier la forme sous laquelle on souhaite obtenir des résultats sur l'incertitude de Y :

✓ On s'intéresse à la loi de probabilité de Y . Pour ce cas, à partir de la loi de probabilité du vecteur X , on en déduit celle de Y . En notant F_X , la distribution de probabilité de X , il faut alors calculer :

$$F_Y(t) = P(Y \leq t) = P(\eta(X) \leq t) = \int_{\eta(x) \leq t} dF_X \tag{Eq. I-3}$$

Cette intégrale n'est malheureusement calculable analytiquement que dans des cas très simples. En général elle est approchée par la distribution empirique de Y , obtenue par échantillonnage (i.e. Monte-Carlo) sur la loi de probabilité X .

✓ En général, on est seulement intéressé par un intervalle de confiance sur Y ou une probabilité de dépassement d'un seuil (analyse de fiabilité). On cherche donc à calculer des probabilités $P(a < Y < b)$ ou $P(Y < 0)$. Dans ce cas, on peut utiliser des méthodes classiques d'estimation de quantiles¹. Si la probabilité $P(Y < 0)$ est faible on peut penser à utiliser des techniques plus sophistiquées telles que FORM (First Order Reliability Method) ou SORM (Second Order Reliability Method) basées sur des développements limités, voir [RAC-01], [SAL-00], [SAL-04], [TSO-08].

I.3.2. Analyse de sensibilité

L'analyse de sensibilité est une étude plus détaillée de celle de l'incertitude. En effet, si l'analyse d'incertitude étudie l'incertitude sur la sortie comme on l'a vu au paragraphe précédent, l'analyse de sensibilité va quant à elle quantifier comment des perturbations sur les entrées du modèle engendrent des perturbations sur la sortie.

On peut regrouper les méthodes d'analyse de sensibilité en trois classes : les méthodes de *screening*, l'analyse de sensibilité locale et l'analyse de sensibilité globale. Les méthodes de *screening* : (présentées et détaillées dans [SAL-00]) analysent qualitativement l'importance des variables d'entrée sur la variabilité de la sortie du modèle. Elles permettent d'établir une hiérarchie au sein des variables d'entrée en fonction de leur influence sur la variabilité de la sortie. L'analyse de sensibilité locale, tout comme l'analyse globale, sont des méthodes d'analyse quantitative, qui permettent en plus d'établir une hiérarchie au sein des variables d'entrée, de donner un ordre de grandeur des écarts au sein de cette hiérarchie.

✚ L'analyse de sensibilité locale étudie comment de petites perturbations autour d'une valeur X_0 (point de fonctionnement) des entrées se répercutent sur la valeur de la sortie. Elle propose d'estimer les dérivées partielles (Eq. I-4) :

$$\frac{\partial \eta}{\partial X_i} \Big|_{X=X_0} \quad \text{Eq. I-4}$$

Ces dérivées expriment l'effet sur la variable de sortie Y, avec la perturbation des variables d'entrée X_i autour d'une valeur nominale X_0 .

¹Points essentiels pris à des intervalles réguliers verticaux d'une fonction de distribution cumulative d'une variable aléatoire

✚ L'analyse de sensibilité globale s'intéresse quant à elle à la variabilité de la sortie du modèle dans son domaine de variation. Elle étudie comment la variabilité des entrées se répercute sur celle de la sortie, en déterminant quelle part de variance de la sortie est due à telles entrées ou tel ensemble d'entrées. Saltelli a présenté l'ensemble des méthodes existantes dans [SAL-00], en les regroupant de la sorte : les méthodes fiabilistes de type FORM et SORM traitant d'analyse de sensibilité pour l'analyse de risques, les méthodes bayésiennes, les méthodes graphiques et enfin les méthodes basées sur l'étude de la variance.

L'analyse de sensibilité est un outil générique et ses objectifs sont multiples [SAL-00] :

- ✓ Modélisation : les paramètres identifiés comme non influents peuvent être ignorés, le modèle s'en trouve avantageusement simplifié.
- ✓ Calibration des facteurs : l'analyse de sensibilité permet de calibrer les paramètres d'entrée compte tenu de contraintes (par exemple de fiabilité) sur la sortie.
- ✓ Prédiction : pour améliorer des prédictions ou bien réduire des risques, l'analyse de sensibilité permet d'identifier les paramètres sur lesquels concentrer les études, améliorer le processus de fabrication, etc.
- ✓ Optimisation : seuls les facteurs influents sont dimensionnés.

I.3.3. L'optimisation, une solution de conception

I.3.3.1. L'optimisation sous contraintes

Durant la conception, le concepteur est amené à rechercher une solution satisfaisant au mieux les besoins fonctionnels, et en même temps viable d'un point de vue économique.

Un problème d'optimisation sous contraintes de dimension n peut se formuler mathématiquement par :

$$\left\{ \begin{array}{l} \text{- minimiser} \quad F(\mathbf{X}) \\ \text{avec} \quad \mathbf{X} = \{x_1, x_2, \dots, x_n\} \in \mathfrak{R}^n \\ \text{et} \quad x_i^{\min} \leq x_i \leq x_i^{\max} \quad i = 1, \dots, n \\ \text{- en respectant} \quad H_j(\mathbf{X}) \leq 0 \quad j = 1, \dots, m \end{array} \right. \quad \text{Eq. I-5}$$

La première formulation comprend une fonction objective F , dont la minimisation rendra compte du respect de ses objectifs. Elle peut être un rendement,

un coût de fabrication etc., ou le poids de la structure, ou un critère physique (force, couple, induction, puissance, etc.) à optimiser.

Les composantes (x_i , $i = 1, \dots, n$) du vecteur \mathbf{X} , qui appartient à un espace de recherche, sont connues sous le nom de variables ou paramètres de conception. Celles-ci peuvent être les dimensions géométriques ou les caractéristiques physiques (propriétés de matériaux, densité de courant...) d'un dispositif.

Le concepteur spécifie aussi le cahier des charges, qui constitue un ensemble de contraintes - $H_j(\mathbf{X})$ associées à la faisabilité du dispositif. Par exemple, elles représentent les limites imposées au problème ou bien pour garantir le bon conditionnement du dispositif (limitation des températures, limitation du courant magnétisant, etc.).

D'une manière générale, les problèmes en ingénierie s'appuyant sur des modèles physiques présentent des grandeurs continues à comportements non linéaires. Ce faisant, les fonction F et H sont des fonctions non linéaires à variables continues. Il est cependant parfois nécessaire de gérer des grandeurs discrètes complexifiant les stratégies d'optimisation. Ce point ne sera pas étudié dans notre travail.

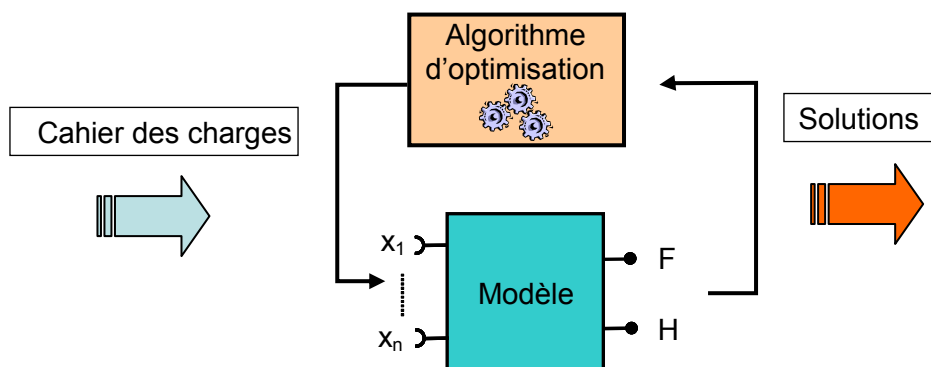


Figure I-7. Processus d'optimisation

La recherche de la solution optimale d'un problème est réalisée au moyen des méthodes d'optimisation. C'est un processus itératif (Figure I-7) de couplage le modèle de dimensionnement du dispositif avec un algorithme d'optimisation. A chaque fois, l'algorithme d'optimisation explore l'espace de recherche pour fournir un jeu de valeurs d'entrée au modèle en regardant si les contraintes imposées par le cahier des charges sont validées, et en analysant la valeur de la fonction objectif correspondante. Le processus itératif s'arrête si la fonction objective arrive à un extremum, et que les contraintes sont respectées.

Le choix des méthodes d'optimisation dépend de la nature du problème (taille du problème, problème mono-objectif ou multiobjectif, etc.). Il dépend également du modèle de dimensionnement du dispositif (modèles analytiques, modèles semi-analytiques, modèles éléments finis). Parmi les algorithmes classiques, on peut distinguer deux grandes familles :

- ✓ Les algorithmes déterministes
- ✓ Les algorithmes stochastiques

Les algorithmes stochastiques s'appuient sur des mécanismes de transition probabilistes et aléatoires qui explorent intelligemment l'espace de recherche. Ces familles d'algorithmes peuvent converger vers un optimum global. Elles ne nécessitent ni point de départ, ni la connaissance du gradient de la fonction objectif pour atteindre une solution optimale. En contrepartie, leur nature aléatoire implique que plusieurs exécutions successives de ces méthodes conduisent à des résultats différents pour une même initialisation du problème d'optimisation. En plus, elles demandent un nombre important d'évaluations du modèle de dimensionnement, donc elles sont très coûteuses en temps d'exécution. Parmi les algorithmes stochastiques les plus employées, on peut distinguer comme dans [COU-02] le *Recuit Simulé*, la *Recherche Tabu* et les méthodes *Evolutionnistes*. Ces dernières représentent une classe d'algorithmes largement utilisée, implémentant des techniques stochastiques comme les *Algorithmes Génétiques*, les *Stratégies d'Evolution*, la *Programmation Evolutionniste*.

Les algorithmes d'optimisation déterministes tiennent leur nom du fait qu'ils évaluent la solution du problème de l'équation Eq. I-5 par un mécanisme reproductible. Les algorithmes les plus performants sont basés sur l'exploitation des gradients du modèle. Ces algorithmes s'appuient sur une direction de recherche fournie par les dérivées de la fonction objective pour converger de manière précise et rapide d'une valeur initiale jusqu'à une valeur optimale. Ces algorithmes ont la capacité de satisfaire un cahier des charges fortement contraint (plusieurs centaines). Cependant, lorsque le problème possède plusieurs optimums, ces algorithmes peuvent converger vers une solution locale. De plus, ils sont très sensibles à la précision des dérivées ; or, le calcul précis des dérivées représente souvent une tâche difficile. Les représentants les plus significatifs de ces algorithmes exploitent le Jacobien, et approximent généralement le Hessien de manière itérative (approche quasi-Newton). Citons la méthode des « points intérieurs » visant des itérations dans l'espace des solutions faisables, ou la méthode que nous utiliserons dans ce travail appelée « *Sequential Quadratic Programming* » (SQP).

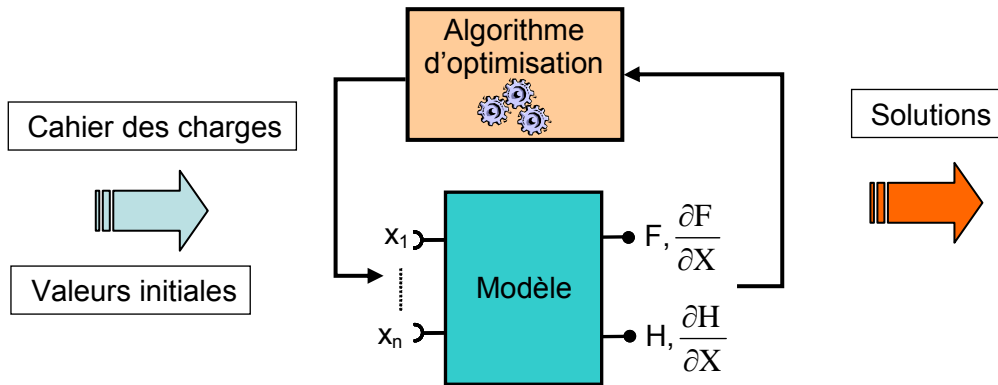


Figure I-8. Processus d'optimisation avec algorithme basé sur le calcul de gradients

L'utilisation des algorithmes d'optimisation basés sur le calcul des gradients est un des points de départ de la problématique de cette thèse. Sachant que ces algorithmes ont une sensibilité élevée aux valeurs des gradients, notre but est d'évaluer ces derniers d'une manière exacte et efficace. Dans le paragraphe I.3.4, nous introduisons des techniques existantes permettant d'évaluer ces gradients, et dans le chapitre III, le développement d'un outil de dérivation automatique des algorithmes décrits en Java sera détaillé.

I.3.3.2. L'optimisation sous contraintes de fiabilité

L'optimisation sous contraintes de fiabilité (RBDO : Reliability-Based Design Optimization) cherche à déterminer une conception optimale, tout en assurant une fiabilité vis à vis des incertitudes [TSO-08].

Le RBDO est formulé comme :

$$\left\{ \begin{array}{l} \text{- minimiser} \quad F(X) \\ \text{avec} \quad X = \{x_1, x_2, \dots, x_n\} \in \mathfrak{R}^n \\ \text{et} \quad x_i^{\min} \leq x_i \leq x_i^{\max} \quad i = 1, \dots, n \\ \text{- en respectant} \quad P(H_j(X) \leq 0) \geq P_{H_j} \quad j = 1, \dots, m \end{array} \right. \quad \text{Eq. I-6}$$

En comparaison avec la formulation du problème d'optimisation classique -Eq. I-5, on trouve : $\{P(H_j(X) \leq 0) \geq P_{H_j}\}$ qui sont les contraintes dites de fiabilité faisant intervenir le vecteur aléatoire X modélisant les incertitudes inhérentes au modèle, et $\{P_{H_j}\}$ qui sont des probabilités de défaillance maximales tolérées.

Bien que les principes de base de RBDO aient été établis depuis plus de 30 ans, leur mise en oeuvre n'est pas encore facile, même pour des modèles simples. La

difficulté réside dans le calcul des contraintes de fiabilité. La situation devient pire quand des modèles numériques (éléments finis par exemple) sont impliqués.

Au cours de ces dernières années, de nombreuses méthodes ont été développées dans les deux domaines : optimisation et fiabilité. Dans [TSO-08], les auteurs ont classées les trois méthodes numériques les plus courantes pour résoudre des problèmes RBDO :

- ✓ Approches à deux niveaux, basées sur l'amélioration de l'approche traditionnelle en double boucle en augmentant l'efficacité de l'analyse de fiabilité.

- ✓ Approches à un niveau, qui visent à résoudre simultanément l'optimisation et les problèmes de fiabilité au sein d'une seule boucle traitant à la fois de conception et de variables aléatoires.

- ✓ Approches découplées, dans lesquelles la contrainte de fiabilité est remplacée par une fonction déterministe équivalente (ou pseudo déterministe) de contrainte, impliquant des simplifications supplémentaires.

Dans le chapitre III, nous exploitons une approche découplée en développant une méthode de propagation d'incertitude pour faire l'optimisation sous contraintes de fiabilité.

I.3.4. Dérivation des modèles de dimensionnement

I.3.4.1. Technique de calcul des gradients

Dans ce paragraphe, nous faisons une synthèse des techniques de calcul des gradients en vue d'une optimisation déterministe ou d'une étude de sensibilité locale.

Considérons une fonction vectorielle décrivant un modèle de dimensionnement :

$$f : x \in \mathcal{R}^n \rightarrow y \in \mathcal{R}^m \quad \text{Eq. I-7}$$

Cette fonction est donnée par une représentation symbolique, ou au moyen d'un programme informatique. Avec n étant le nombre des variables indépendantes (entrées) et m le nombre des variables dépendantes (sorties), donc pour lesquelles les dérivées partielles sont à calculer. La dérivée directionnelle est définie par :

$$\dot{y} = \lim_{h \rightarrow 0} \frac{f(x + h \cdot \dot{x}) - f(x)}{h} \quad \text{Eq. I-8}$$

où $\dot{x} \in \mathcal{R}^n$ est la direction de calcul des dérivées.

a) Approximation par différences finies

La méthode de différences finies (DF) vient de la définition de la dérivée (Eq. I-8).

$$\dot{y} = \frac{f(x+h.x) - f(x)}{h} \quad \text{Eq. I-9}$$

La précision de cette approximation dépend fortement du pas de dérivation h . Pour un pas trop grand, la précision de la dérivée peut être gravement affectée par l'erreur de la méthode. A contrario, pour un pas trop petit, le principe de cette méthode peut être à l'origine de problème d'instabilités numériques engendrées par les erreurs de troncature.

Cette technique est très facile à mettre en oeuvre mais elle est très difficile à régler, c'est-à-dire à trouver le bon compromis pour le pas de dérivation h [PRE-91]. Des techniques de calcul par pas adaptatifs sont envisageables [FMI], mais ces calculs ne restent qu'approximatifs et très coûteux.

b) Calcul symbolique

Le calcul symbolique (ou calcul formel) des dérivées est implémenté dans des logiciels comme Maple [KOF], Mathematica [WOL] ou Macsyma [PET], etc. Ceux-ci supposent que la fonction que l'on cherche à dériver admet une représentation symbolique au moyen de formules mathématiques. On obtient alors une représentation, également symbolique, de la fonction dérivée. Ce point de vue peut être considéré comme une automatisation de ce qu'un être humain a appris à faire sur une feuille de papier. Notons que des logiciels comme Maple, Mathematica ou Macsyma vont bien au delà de la différentiation de formules mathématiques et offrent toute une panoplie de manipulation symboliques telles que la simplification d'expression, la résolution de fonctions (simples). Par contre, ils ne visent pas à dériver des fonctions représentées par des programmes informatiques.

c) Dérivation automatique de code

La Dérivation Automatique (DA) exploite le processus de la compilation du code source d'un programme informatique, et fait usage de la représentation intermédiaire des fonctions mises en oeuvre dans différents langages de programmation. La dérivée de chaque opération élémentaire peut être obtenue de manière simple et peut être combinée selon la règle de la chaîne du calcul différentiel afin d'obtenir la dérivée de la fonction initiale. Cette technique fournit un moyen efficace pour calculer des gradients en réduisant énormément l'effort d'implémentation.

Sur la base de ces principes, la DA est implémenté dans certain nombre d'outils pour les langages de programmation comme Fortran, C, C++ ou Matlab, dont la plupart sont listés dans la base de données actualisée régulièrement [AUT].

I.3.4.2. Dérivation de modèle de dimensionnement dans CadesGenerator

Pour décrire le modèle de dimensionnement, CADES utilise le langage de modélisation *sml*¹ qui a été développé aux cours de ces dernières années au G2Elab à travers les travaux de thèse de E. Atienza [ATI-03], B. Delinchant [DEL-03], L. Allain [ALL-03], V. Fischer [FIS-04], D. Magot [MAG-04], B. Du Peloux [DUP-06], H. L. Rakotoarison [RAK-07], D. Duret [DUR-07] et P. Enciu [ENC-09-2].

Le *CadesGenerator* est le point d'entrée du framework CADES, qui génère un composant de calcul (ICAr²) à partir d'un modèle de dimensionnement (*sml*).

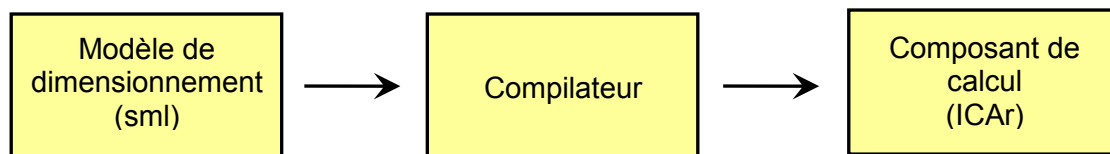


Figure I-9. Génération du composant de calcul de CADES

Le compilateur spécialisé pour le langage *sml* génère un programme cible, encapsulant, conformément à la norme de CADES (ICAr), le modèle mathématique de dimensionnement et ses gradients. Le programme cible, se présentant comme un composant de calcul, est exécutable par une machine virtuelle Java.

Avant notre intervention, il existait deux compilateurs du langage *sml* :

- ✚ *JavaGenerator* : ce compilateur, écrit en Java, utilise le calcul symbolique des dérivées et fonctionne seulement pour des modèles analytiques scalaires explicites. Pour la modélisation semi analytique, le concepteur fournit des fonctions externes décrites dans un langage externe (Java), ainsi que le code de calcul du Jacobien de ces fonctions. C'est une tâche difficile qui demande de gros efforts d'implémentation. Avec ces inconvénients, ce compilateur ne peut être appliqué que sur un nombre restreint d'applications.

¹ *sml* : System Modelling Language

²ICAr : Interfaces for Component Architecture – Le nom de la norme du composant logiciel de CADES

✚ *AdolCGenerator* : ce compilateur applique le dérivateur automatique ADOL-C [WAL] en C/C++ pour dériver des modèles de dimensionnement vectorisés. Comme l'architecture du composant de calcul (ICAr) est implémentée en Java, il faut passer par l'interface JNI¹ pour faire le lien avec le noyau (en C).

L'environnement CADES offre une grande souplesse aux développeurs de modèles, avec un niveau « *experts* » permettant d'étendre les capacités de modélisation via la notion des fonctions externes. Ces fonctions peuvent être décrites en C ou en Java, mais doivent fournir en complément, le Jacobien du modèle. Lorsque les interactions entre les deux langages sont importantes, les problèmes de mémoire et de temps d'exécution ne sont pas négligeables. Pour remédier à ce problème, il serait nécessaire de générer automatiquement le calcul des gradients de modèles semi-analytiques, en Java.

I.4. Conclusions

Dans ce chapitre, nous avons abordé les généralités sur les MEMS/NEMS. Tout d'abord, des principes d'actionnement dans les MEMS et ceux, plus particuliers, du nano commutateur magnétique, ont été présentés. Nous avons présenté ensuite les outils existants de modélisation des MEMS pour mettre en évidence la nécessité de développer un outil spécifique dédié au nano commutateur magnétique. Nous avons aussi introduit dans ce chapitre les outils d'analyse comme l'analyse d'incertitude, l'analyse de sensibilité, l'optimisation sous contraintes, l'optimisation sous contraintes de fiabilité. Nous avons détaillé enfin les techniques d'évaluation du gradient et l'état de l'art du framework CADES pour montrer ses potentialités dans le cas de notre problème et ses lacunes que nous avons essayé de combler.

¹ JNI : *Java Native Interface*

Chapitre II

Modélisation magnéto-mécanique du nano commutateur

SOMMAIRE

Chapitre II

Modélisation magnéto-mécanique du nano commutateur	35
II.1. Modèle magnétique	37
II.1.1. Modèle magnétique dans MacMMems.....	37
II.1.1.1. Hypothèses	38
II.1.1.2. Les conducteurs.....	39
II.1.1.3. Les aimants	39
II.1.2. Modèles magnétiques spécifiques (FM/AF).....	43
II.2. Modèle de force de Casimir.....	44
II.3. Modèle mécanique	44
II.3.1. Hypothèses.....	45
II.3.2. Réduction de la poutre de 3D en 1D	46
II.3.2.1. Cas d'une poutre parallélépipédique	46
II.3.2.2. Cas d'une poutre constituée de plusieurs parallélépipèdes de même longueur.....	47
II.3.2.3. Cas d'une poutre constituée de plusieurs parallélépipèdes	49
II.3.3. Résolution de l'équation d'Euler - Bernoulli.....	49
II.3.4. Analyse du contact mécanique.....	50
II.3.5. Validation par éléments finis.....	55
II.4. Couplage magnéto-mécanique	57
II.4.1. Notions de couplage multi-physiques	57
II.4.2. Couplage magnéto-mécanique.....	57
II.5. Introduction du composant déformable dans MacMMems.....	59
II.5.1. Interface Homme / Machine (IHM)	59
II.5.2. Visualisation de la déformation	62
II.6. Conclusions.....	63

Résumé du chapitre

Ce chapitre porte sur la modélisation du couplage magnéto-mécanique. Un modèle semi analytique dédié au nano commutateur magnétique y est développé. Tout d'abord, les modèles magnétiques existants dans MacMMems seront présentés. L'empilement spécifique FM/AF, utilisé dans le nano commutateur, sera considéré comme un aimant permanent en régime statique. Ensuite, nous présenterons le modèle mécanique développé pour le calcul de la déformation de la poutre avec l'analyse du contact mécanique. La validation de ce nouveau modèle semi analytique par une méthode éléments finis sera aussi présentée. Nous finirons enfin par le couplage magnéto-mécanique et la prise en compte des forces de Casimir dans le logiciel MacMMems.

II.1. Modèle magnétique

Dans le cadre de notre application, nous ne considérons pas d'effets magnétiques microscopiques [BIL-09], [VIA-08], ce qui nous a conduit à utiliser des méthodes classiques de modélisation magnétique macroscopiques telles que celle développées dans l'outil MacMMems.

II.1.1. Modèle magnétique dans MacMMems

Comme nous l'avons présenté brièvement dans la section I.2.2.2 du premier chapitre, le logiciel MacMMems est un générateur de macro modèles (analytique et semi analytique) pour les MEMS magnétiques. Ces modèles sont exploitables dans le framework CADES.

Afin de faciliter leur fabrication, les MEMS sont généralement réalisés avec des formes géométriques simples. Leurs topologies peuvent être décomposés par :

- ✓ Des parallélépipèdes
- ✓ Des secteurs cylindriques (pour un angle de 2π , on a des couronnes, pour un rayon intérieur nul, on a des disques ou des cylindres)
- ✓ Des sphères (exploitées essentiellement en bio-électromagnétisme [CHE-07])

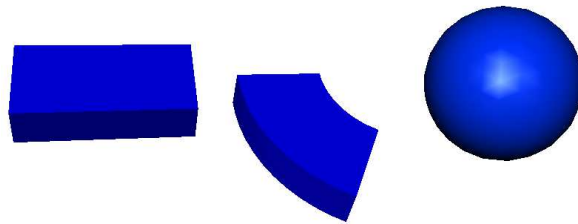


Figure II-1. Les formes de bases dans les structures des MEMS magnétique

De ce fait, on peut décomposer un MEMS en plusieurs entités et modéliser chaque partie par des modèles analytiques. Pour modéliser le système magnétique global, MacMMems utilise le principe de modélisation des composants en interaction.

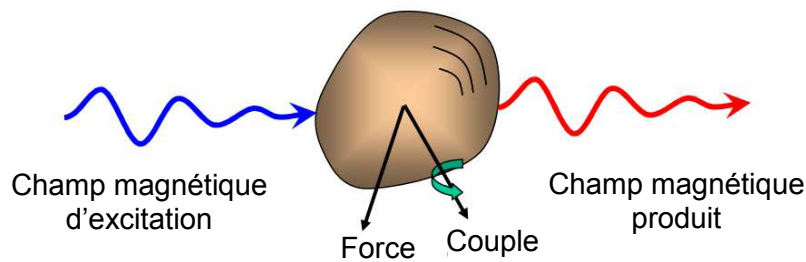


Figure II-2. Représentation d'un composant MEMS magnétique dans MacMMems [RAK-07]

En présence d'un champ magnétique d'excitation, un composant magnétique est soumis à une force et/ou un couple. Ce composant peut être aussi amené à s'aimanter et produire lui-même un champ magnétique qui est une fonction ou non du champ exciteur.

II.1.1.1. Hypothèses

Afin d'obtenir un modèle analytique, permettant de modéliser des composants de MEMS magnétiques, MacMMems pose des hypothèses simplificatrices suivantes [RAK-07] :

- ✓ L'étude est statique ou quasi statique.
- ✓ Les densités de courant dans les conducteurs sont uniformes.
- ✓ Les aimants permanents sont rigides. C'est-à-dire que leur aimantation est insensible au champ externe, et à leur propre champ démagnétisant. Cette hypothèse est valide pour quelques types d'aimants couramment utilisés dans des MEMS (NdFeBr).
- ✓ Pour les matériaux ferromagnétiques, le phénomène d'hystérésis est négligé.
- ✓ Les déformations des composants magnétiques ne sont pas prises en compte.

Grâce à l'introduction du couplage magnéto-mécanique du composant déformable (voir la section II.5), cette dernière hypothèse n'est plus nécessaire aujourd'hui.

Nous allons maintenant détailler comment sont modélisés les conducteurs et les aimants permanents, qui peuvent être utilisés dans le couplage magnéto-mécanique du nano commutateur.

II.1.1.2. Les conducteurs

Les conducteurs sont les entités qui permettent de commander les dispositifs électromagnétiques. Comme nous allons voir dans la section II.1.2, le principe de retournement de l'aimantation de l'empilement spécifique ferromagnétique/anti-ferromagnétique (FM/AF), est d'utiliser des conducteurs pour chauffer, et ainsi désaimanter et fournir un champ magnétique pour ré-aimanter.

MacMMems utilise la loi de *Biot-Savart* pour calculer le champ magnétique créé par le conducteur en tout point de l'espace :

$$\vec{H}(\vec{r}) = \frac{1}{4\pi} \left[\iiint_v \frac{\vec{j} \times (\vec{r} - \vec{r}')}{|\vec{r} - \vec{r}'|^3} dv \right] \quad \text{Eq. II-1}$$

où le vecteur \vec{r} représente la position du point de calcul, et le vecteur \vec{r}' définit la position de l'élément volumique dv . \vec{j} correspond à la densité surfacique de courant.

Sur les conducteurs de topologies illustrées sur la Figure II-1 (sauf la sphère), l'intégration de l'équation Eq. II-1 peut être résolue symboliquement complètement ou seulement partiellement, conduisant à des modèles analytiques ou semi-analytiques (voir [RAK-07]).

II.1.1.3. Les aimants

a) Le champ produit

Avec l'hypothèse de rigidité, un aimant est caractérisé par son aimantation - M .

L'approche Coulombienne représente un aimant par des charges magnétiques fictives, surfaciques et/ou volumiques.

$$\sigma_s = \vec{M} \cdot \vec{n} \quad \text{Eq. II-2}$$

$$\sigma_v = -\text{div}(\vec{M}) \quad \text{Eq. II-3}$$

Dans l'approche Coulombienne, par analogie à l'électrostatique, le champ magnétique créé par l'aimant s'exprime par :

$$\vec{H}(\vec{r}) = \iiint_v \frac{\sigma_v \cdot (\vec{r} - \vec{r}')}{|\vec{r} - \vec{r}'|^3} dv + \iint_s \frac{\sigma_s \cdot (\vec{r} - \vec{r}')}{|\vec{r} - \vec{r}'|^3} ds \quad \text{Eq. II-4}$$

où le vecteur \vec{r} représente la position du point de calcul, et le vecteur \vec{r}' définit la position de l'élément volumique dv et/ou l'élément surfacique ds .

Les formules analytiques, obtenues par la résolution symbolique de l'équation Eq. II-4 sur les aimants de topologie décrite sur la Figure II-1, sont détaillées dans [RAK-07].

b) La force

Pour des configurations spécifiques, telles que les interactions entre les aimants parallélépipédiques, la force peut être calculée analytiquement [AKO-84]. Dans MacMMems, les interactions n'étant pas définies par avance, et pouvant mettre en jeu un nombre de composants et de topologies important, la force sur l'aimant, en présence du champ extérieure \vec{H}_{ext} est calculée numériquement par l'intégration :

$$\vec{F} = \iint_S \sigma_s \cdot \vec{H}_{ext} ds + \iiint_V \sigma_v \cdot \vec{H}_{ext} dv \quad \text{Eq. II-5}$$

c) Le couple

Le couple exercé sur l'aimant peut être décomposé en 2 composantes :

✓ La composante qui provient de la force est une fonction du point du pivot. Cette composante se calcule par le produit vectoriel de la force et du bras de levier :

$$\vec{\Gamma}_{force} = \vec{AP} \times \vec{F} \quad \text{Eq. II-6}$$

où A est le point du pivot, P est le point d'application de la force.

En fait, on ne sait pas où se trouve ce point P, mais en faisant l'hypothèse d'un champ lointain, on prendra le centre géométrique de l'aimant.

✓ La composante qui provient du produit vectoriel entre l'aimantation et le champ magnétique externe. Son existence s'explique par le fait que dans un champ uniforme, l'aimant ne subit aucune force, mais qu'il peut subir un couple. Cette composante de couple est appelée « couple magnétique ». Ce couple magnétique est indépendant de la force liée au point du pivot. Le couple magnétique se calcule par :

$$\vec{\Gamma}_{magnétique} = \iiint_V \vec{M} \times \vec{H}_{ext} dv \quad \text{Eq. II-7}$$

La première composante du couple est approximée avec l'hypothèse de champ lointain, donc avec la même hypothèse, le couple magnétique peut être calculé approximativement par :

$$\vec{\Gamma}_{magnétique} = V \cdot (\vec{M} \times \vec{H}_{ext}) \quad \text{Eq. II-8}$$

où V est le volume de l'aimant. \vec{M} et \vec{H}_{ext} sont pris au centre de l'aimant.

Dans MacMMems, avant notre intervention, le couple calculé sur un aimant correspondait au couple total. Avec l'approche Coulombienne, l'aimant est bien équivalent à des charges magnétiques fictives, surfaciques σ_s et/ou volumiques σ_v , le couple total est calculé par :

$$\vec{\Gamma}_{\text{total}} = \iint_S \vec{AP}_S \times (\sigma_s \cdot \vec{H}_{\text{ext}}) ds + \iiint_V \vec{AP}_V \times (\sigma_v \cdot \vec{H}_{\text{ext}}) dv \quad \text{Eq. II-9}$$

où A est le point du pivot, P_s est la position de l'élément surfacique ds , P_v est la position de l'élément volumique dv .

Considérons un exemple simple d'interaction entre deux aimants cubiques (de taille $1 \mu\text{m}$) comme le montre la Figure II-3.

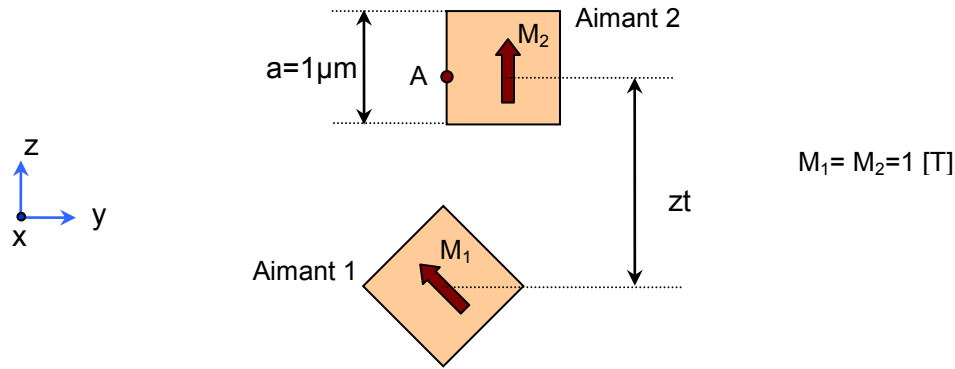


Figure II-3. Exemple d'interaction entre deux aimants

La Figure II-4 montre la variation du couple magnétique sur un aimant en fonction de distance entre eux (zt). On peut observer que si zt est faible par rapport à la taille de l'aimant (l'hypothèse de champ lointain n'est plus valable) il y a un petit écart entre l'approximation par Eq. II-8 et la référence calculée par l'intégration en trois dimensions Eq. II-7.

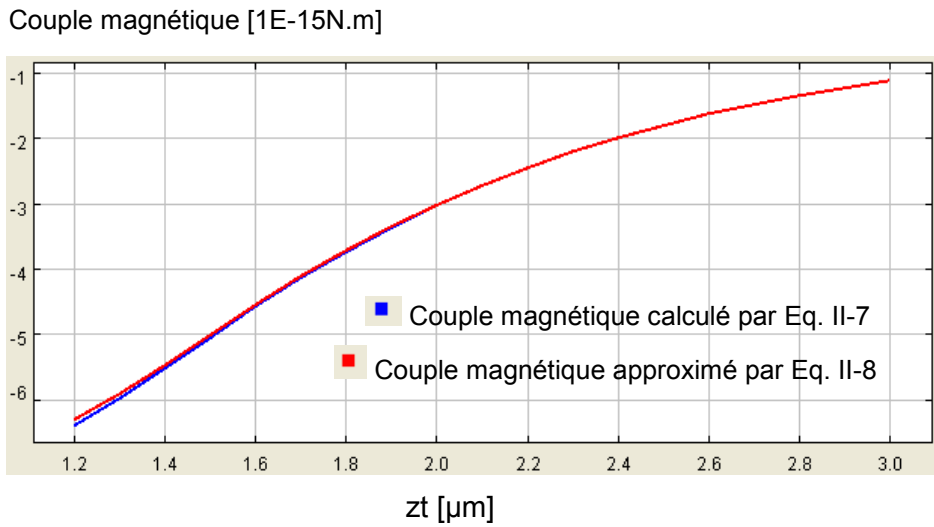


Figure II-4. Couple magnétique

Afin de valider l'approche de décomposition des couples (en deux composantes) ci-dessus, la Figure II-5 trace la variation du couple total sur un aimant en fonction de la distance entre les deux. On peut aussi observer un écart entre les deux courbes si zt est de petite valeur. Cet écart peut être expliqué par des approximations sur les deux composantes calculées par Eq. II-6 et Eq. II-8.

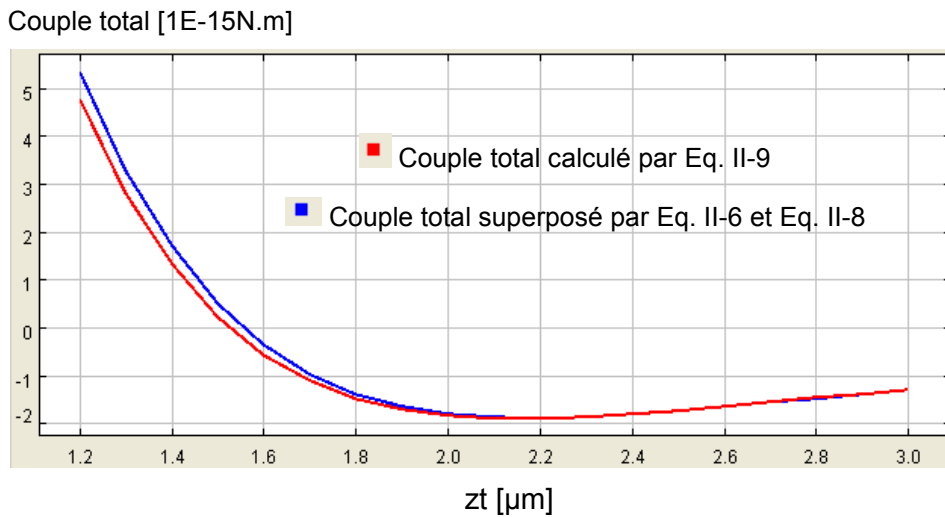


Figure II-5. Couple total

Dans le couplage magnéto-mécanique, comme les entrées du modèle mécanique sont des forces et couples ponctuels (voir la section II.3), la force et le couple magnétique sont utilisés. Pour éviter l'intégration en trois dimensions Eq. II-7, on peut approximer le couple magnétique par Eq. II-8.

II.1.2. Modèles magnétiques spécifiques (FM/AF)

Dans cette section, le principe de retournement de l'aimantation de l'empilement spécifique ferromagnétique/antiferromagnétique (FM/AF), utilisé dans le nano commutateur magnétique, est présenté brièvement. Pour plus de détails sur ce principe, le lecteur peut consulter [BIL-09], [VIA-08].

Cet empilement est composé d'une couche de matériau ferromagnétique (NiFe, CoFe, etc.) et d'une couche d'un matériau antiferromagnétique (IrMn, PtMn, etc.).

Si la couche AF est chauffée, lorsque la température dépasse sa température de Néel ($T_{\text{Néel}}$), le matériau devient paramagnétique. Le couplage d'échange, qui maintenait rigide l'aimantation de la couche FM disparaît. La couche FM retrouve alors son caractère doux, et son aimantation s'aligne suivant le champ extérieur.

Si la température de la couche AF est abaissée en dessous de $T_{\text{Néel}}$ et que le champ extérieur est maintenu, le couplage d'échange réapparaît. La couche FM garde le sens de l'aimantation imposée par le champ extérieur même si ce champ disparaît [PRE-04], [PRE-07].

La Figure II-6, extrait de [PRE-07], représente le mécanisme d'écriture de la couche FM/AF. A température ambiante, si le champ extérieur est nul, le cycle d'hystérésis de l'aimantation est décalé de H_{ex} (champ d'échange – *exchange field*). L'aimantation rémanente de la couche FM est la valeur de l'aimantation à saturation (ici négative, voir Figure II-6- (a)). Lorsque l'empilement est chauffé jusqu'à la température de Néel (dans la pratique on parle plutôt de la température de blocage), H_{ex} s'annule, le cycle se décale vers l'origine (voir Figure II-6- (b)). Si dans le même temps, on applique un champ magnétique extérieur H_{sw} (**switching field**), le cycle se décale à nouveau de H_{ex} mais avec un signe opposé. Lorsque la température s'abaisse, l'empilement conserve une aimantation rémanente non nulle de sens opposé en l'absence de champ magnétique extérieur (voir Figure II-6- (c)).

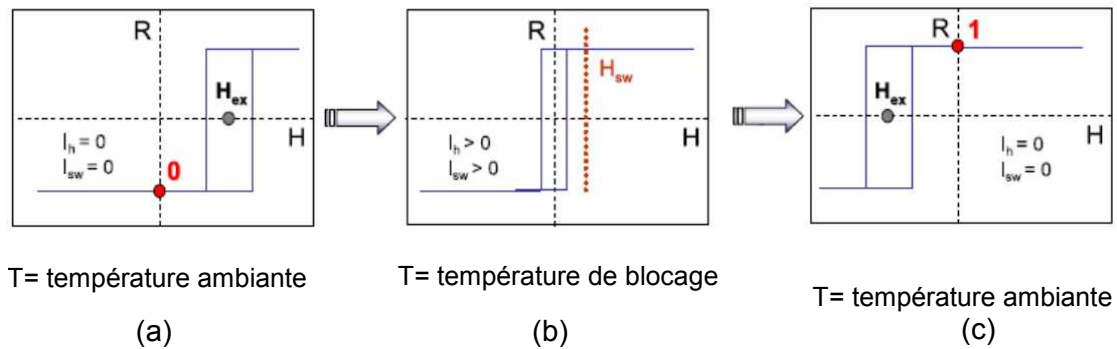


Figure II-6. Mécanisme de retournement d'aimantation de la couche FM/AF [PRE-07]

Dans le processus de retournement de l'aimantation ci-dessus, le conducteur a deux rôles : chauffer et fournir le champ externe.

En régime statique, cet empilement FM/AF se comporte globalement comme des aimants permanents dont l'aimantation vaudrait quasiment celle à saturation. Nous pouvons donc modéliser ces objets par l'approche Coulombienne de MacMMems, en considérant que le volume de l'aimant correspond au volume de la couche ferromagnétique.

II.2. Modèle de force de Casimir

La force de Casimir, prédite par le physicien néerlandais Hendrik Casimir en 1948, est une force attractive entre deux plaques parallèles conductrices et non chargées. Cet effet, dû aux fluctuations quantiques du vide existe également pour d'autres géométries d'électrodes [DUP-03].

Dans le cadre du projet MONACO, le partenaire LKB (Laboratoire Kastler Brossel) a développé un modèle numérique permettant de calculer l'énergie de Casimir donc en déduire la force de Casimir entre deux objets plans comportant des corrugations. Ce code a été validé par des mesures expérimentales [BAO-10].

Afin d'intégrer dans MacMMems, les modèles analytiques de force de Casimir développé dans [DUP-03], [BAL-03], [LAM-03] sont envisagés. Ces modèles expriment la force de Casimir par unité de surface.

II.3. Modèle mécanique

De façon traditionnelle, le calcul de déformation de poutres fait partie du domaine de la résistance des matériaux (RDM). Cette discipline, longtemps développée en tant que telle, a permis pendant longtemps de calculer analytiquement des ponts,

des ouvrages d'art divers. Les mêmes calculs sont aussi effectués de façon numérique, au moyen de codes de calcul par éléments finis.

Pour modéliser le nano commutateur magnétique afin d'évaluer la force et la longueur de contact, nous avons besoin du modèle de déformation de la poutre avec analyse du contact mécanique. Au cours de notre recherche bibliographique, nous n'avons trouvé que des solutions exploitant des modèles éléments finis physiques comme Ansys® [ANS] ou Comsol® [COM]. Or, ce type de modélisation est inadapté à notre contexte d'optimisation sous contraintes ou d'optimisation sous contraintes de fiabilité qui nécessite un modèle extrêmement léger.

Dans cette section, le modèle semi analytique de calcul déformation de la poutre avec analyse du contact mécanique que nous avons développé va être présenté. Ce modèle a les paramètres d'entrées/sorties suivants (Figure II-7).

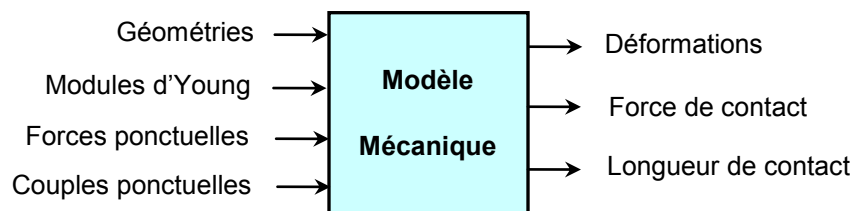


Figure II-7. Entrées/sorties du modèle mécanique de calcul déformation de la poutre avec analyse du contact mécanique

II.3.1. Hypothèses

Le modèle est basé sur quatre hypothèses de simplification suivantes :

- ✚ Les matériaux sont linéaires et isotropes.
- ✚ Hypothèse de Bernoulli (hypothèse de petites déformations) : les sections droites, planes et perpendiculaires à la ligne moyenne, restent planes et perpendiculaires à la ligne moyenne après déformation [TIM-49]. Cette hypothèse n'est valable que pour des poutres longues, c'est-à-dire que la dimension des sections et la dimension de déformation doivent être petites devant la longueur de la poutre. Dans le cas du nano commutateur magnétique, cette hypothèse est bien valable car la poutre a une longueur très supérieure ($\sim 3\mu\text{m}$) à la dimension de déformation ($\sim 100\text{nm}$). Cette hypothèse est valable pour les MEMS de manière générale.

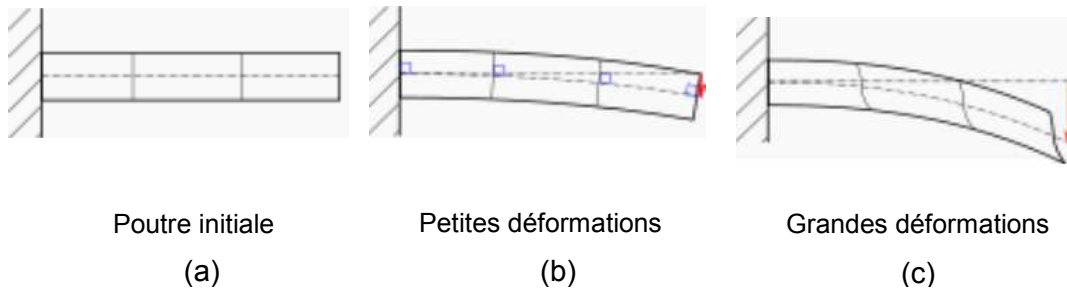


Figure II-8. Illustration de l'hypothèse de Bernoulli

- ✚ Une seule direction de déformation est considérée (déformation principale - verticale) : on néglige toutes les déformations secondaires, dans le plan horizontal, ainsi que les torsions.

- ✚ Les frottements au contact sont négligés.

II.3.2. Réduction de la poutre de 3D en 1D

Compte tenu des formes géométriques simples de MEMS/NEMS en général et en particulier pour le nano commutateur magnétique, la poutre peut être décomposée en parallélépipèdes. De plus, avec des hypothèses simplificatrices mentionnées dans la section II.3.1 ci-dessus, la poutre en trois dimensions (3D) peut être réduite en une dimension (1D).

Cette réduction consiste à déterminer le EI_{eq} (produit équivalent E.I) de la poutre. Avec :

- ✓ E : module d'Young ou module d'élasticité ou encore module de traction. C'est une constante d'un matériau élastique isotrope qui relie la contrainte de traction (ou de compression) et la déformation.

- ✓ I : moment quadratique. C'est une grandeur qui caractérise la géométrie d'une section et se définit par rapport à un axe ou un point.

II.3.2.1. Cas d'une poutre parallélépipédique

Pour une poutre parallélépipédique, la réduction de 3D en 1D est illustrée sur la Figure II-9. Le moment quadratique de section rectangulaire, pour la direction de déformation en z, est calculé par :

$$I_1 = \frac{y_1 \cdot z_1^3}{12} \quad \text{Eq. II-10}$$

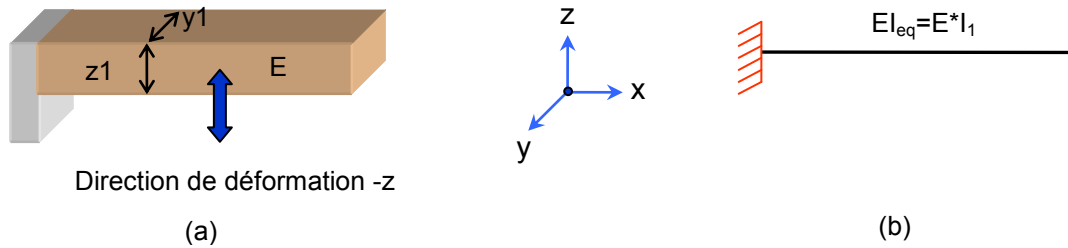


Figure II-9. (a) Poutre parallélépipédique et (b) sa poutre équivalente 1D

II.3.2.2. Cas d'une poutre constituée de plusieurs parallélépipèdes de même longueur

Pour une poutre inhomogène, qui se compose de plusieurs parallélépipèdes de modules d'Young différents, mais de même longueur (en x), la réduction de 3D en 1D comprend quatre étapes :

a) Etape 1 : Calcul des profondeurs équivalentes

Cette poutre inhomogène de modules d'Young différents peut être remplacée par celle homogène de module d'Young du premier parallélépipède. Pour cela, des profondeurs équivalentes sont calculées pour chaque parallélépipède par :

$$y_{eq-i} = y_i \cdot \frac{E_i}{E_1} \quad \text{Eq. II-11}$$

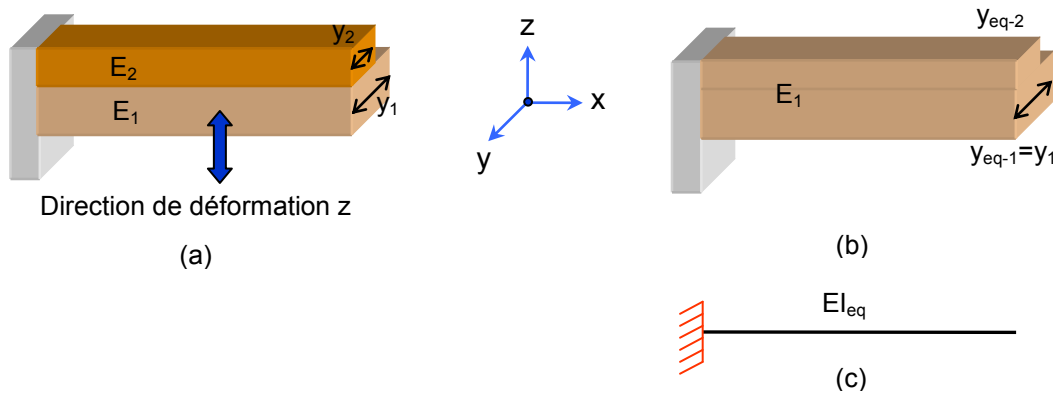


Figure II-10. (a) Poutre inhomogène de plusieurs parallélépipèdes de même longueur, (b) Poutre homogène équivalente et (c) poutre équivalente 1D

b) Etape 2 : Calcul de l'axe neutre

Si l'on part d'un état initial où la poutre n'est pas déformée, la position de l'axe neutre z_G correspond à son centre géométrique. Cette position est trouvée en

déterminant celle du barycentre G de la poutre, c'est-à-dire en résolvant l'équation suivante :

$$\sum_{i=1}^n S_i \cdot \overrightarrow{GP_i} = 0 \quad \text{Eq. II-12}$$

où S_i est la surface de la section i , P_i est le barycentre de la couche i .

La position de l'axe neutre z_G se calcule par :

$$z_G = \frac{\sum_{i=1}^n z_{p-i} \cdot y_{eq-i} \cdot z_i}{\sum_{i=1}^n y_{eq-i} \cdot z_i} \quad \text{Eq. II-13}$$

où y_{eq-i} est la profondeur équivalente, z_i est la hauteur et z_{p-i} est la position du centre (en z) du parallélépipède i .

c) Etape 3 : Calcul du moment quadratique de chaque parallélépipède par rapport à l'axe neutre

Le moment quadratique de chaque parallélépipède par rapport à l'axe neutre est déterminé par le théorème de transport de *Huygens* :

$$I_{neutre-i} = I_i + A_i \cdot d_i^2 \quad \text{Eq. II-14}$$

où :

✓ $I_i = \frac{y_{eq-i} \cdot z_i^3}{12}$ est le moment quadratique du parallélépipédique i par

rapport à son centre.

✓ $d_i = |z_G - z_{p-i}|$ est la distance entre l'axe neutre et la position du centre du

parallélépipède i .

✓ $S_i = y_{eq-i} \cdot z_i$ est la surface de la section du parallélépipède i .

d) Etape 4 : Calcul EI_{eq} de la poutre

Finalement, le EI_{eq} de la poutre 1D équivalente est calculé par :

$$EI_{eq} = E_1 \cdot \sum_{i=1}^n I_{neutre-i} \quad \text{Eq. II-15}$$

II.3.2.3. Cas d'une poutre constituée de plusieurs parallélépipèdes

Pour la poutre inhomogène qui se compose de plusieurs parallélépipèdes de modules d'Young différents comme le montre la Figure II-11, on peut la décomposer en plusieurs sections et appliquer les cas précédents II.3.2.1. et/ou II.3.2.2.

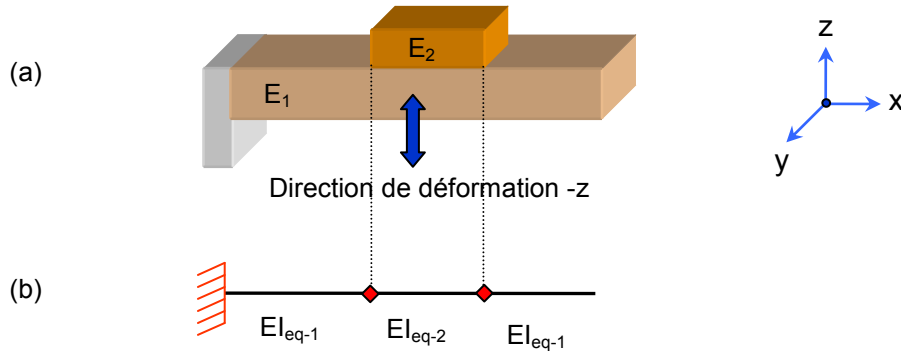


Figure II-11. (a) Poutre de plusieurs parallélépipèdes et (b) sa poutre équivalente 1D

II.3.3. Résolution de l'équation d'Euler - Bernoulli

Une fois que la poutre est ramenée en 1D, l'équation à résoudre pour calculer la déformation est l'équation différentielle Euler-Bernoulli :

$$EI_{eq} \cdot \frac{\partial^2 z}{\partial x^2} = M \quad \text{Eq. II-16}$$

où :

- ✓ $z(x)$ est la déformation de la poutre en position x
- ✓ M est le moment de flexion en position x

Le principe de superposition est utilisé compte tenu de la déformation élastique linéaire : La déformation totale est égale à la somme des déformations créées par chaque force et chaque couple :

$$z_{total}(x) = \sum_{toutes_forces_et_couples} z_i(x) \quad \text{Eq. II-17}$$

De plus, la dérivée de la déformation (par rapport à x) est aussi calculée pour l'analyse du contact mécanique :

$$\frac{\partial z_{total}(x)}{\partial x} = \sum_{toutes_forces_et_couples} \frac{\partial z_i(x)}{\partial x} \quad \text{Eq. II-18}$$

Pour chaque force ou chaque couple, le moment de flexion en x est de la forme :

$$M = C_1 \cdot x + C_2 \quad \text{Eq. II-19}$$

Les solutions de l'équation d'Euler-Bernoulli (Eq. II-16) sont :

$$z_1(x) = C_1 \cdot \frac{x^3}{6} + C_2 \cdot \frac{x^2}{2} + C_3 \cdot x + C_4$$

$$\frac{\partial z(x)_i}{\partial x} = C_1 \cdot \frac{x^2}{2} + C_2 \cdot x + C_3 \quad \text{Eq. II-20}$$

Dans des équations Eq. II-19, Eq. II-20, C_1 , C_2 , C_3 , C_4 sont des constantes déterminées par des conditions aux limites et conditions de continuité de passage entre des différentes sections de la poutre de EI_{eq} différents.

Les conditions aux limites dépendent du type de poutre, comme illustré par la Figure II-12. Nous verrons plus loin que, pour la prise en compte du contact, nous exploiterons ces trois types.

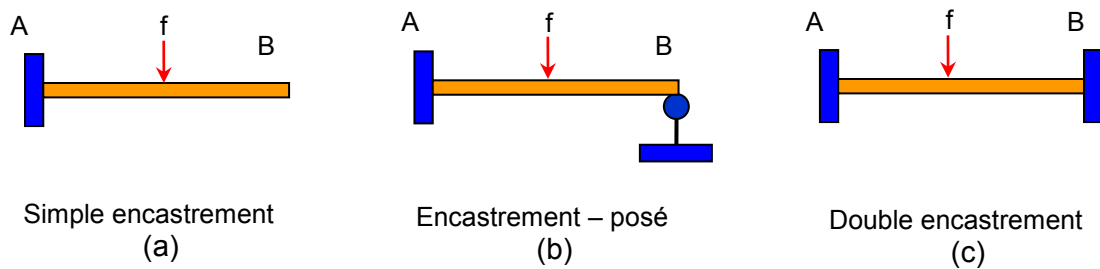


Figure II-12. Trois types de poutre, classés en fonction de condition de support aux extrémités

- ✓ Pour l'encastrement en A, les conditions aux limites sont :

$$z_1(x)_A = 0, \quad \frac{\partial z_1(x)_A}{\partial x} = 0 \quad \text{Eq. II-21}$$

- ✓ Pour le posé en B, la condition à la limite est :

$$z_1(x)_B = 0 \quad \text{Eq. II-22}$$

- ✓ Pour l'encastrement en B, les conditions aux limites sont équivalentes à celles de l'encastrement en A.

II.3.4. Analyse du contact mécanique

L'analyse du contact mécanique est réalisée par une décomposition en trois étapes, qui se succèdent comme le montre la Figure II-13 et dont les effets se superposent comme décrit dans l'algorithme de la Figure II-14.

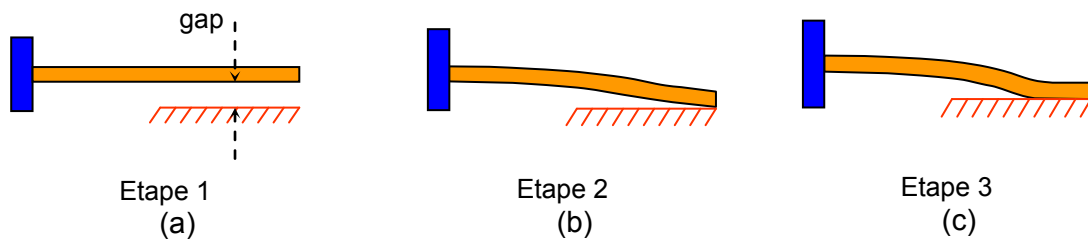


Figure II-13. Trois étapes de l'analyse de contact mécanique

a) Etape 1

Dans cette étape, nous calculons a_0 , qui est la proportion des forces et des couples totaux nécessaires pour que la poutre touche le contact :

$$a_0 \cdot z_{1-B} = \text{gap} \quad \text{Eq. II-23}$$

où z_{1-B} est la déformation à l'extrémité de la poutre (en B) créée par toutes les forces et tous les couples, poutre de type simple encastrement (Figure II-13- (a)).

Si $a_0=1$, la poutre est en contact ponctuel sans force d'appui.

b) Etape 2

Dans cette étape, l'idée est ici d'atteindre la limite entre le contact ponctuel et le contact surfacique. Nous calculons pour cela a_1 , avec $(a_0 + a_1)$ qui correspond à la proportion des forces et des couples totaux pour que la dérivée de la déformation à l'extrémité de la poutre (en B) soit nulle.

$$a_0 \cdot \frac{\partial z_{1-B}}{\partial x} + a_1 \cdot \frac{\partial z_{2-B}}{\partial x} = 0 \quad \text{Eq. II-24}$$

où :

✓ $\frac{\partial z_{1-B}}{\partial x}$ est la dérivée de la déformation à l'extrémité de la poutre (en B)

créée par toutes les forces et tous les couples, poutre de type simple encastrement (Figure II-13- (a)).

✓ $\frac{\partial z_{2-B}}{\partial x}$ est la dérivée de la déformation à l'extrémité de la poutre (en B)

créée par toutes les forces et tous les couples, poutre de type encastrement-posé (Figure II-13- (b)).

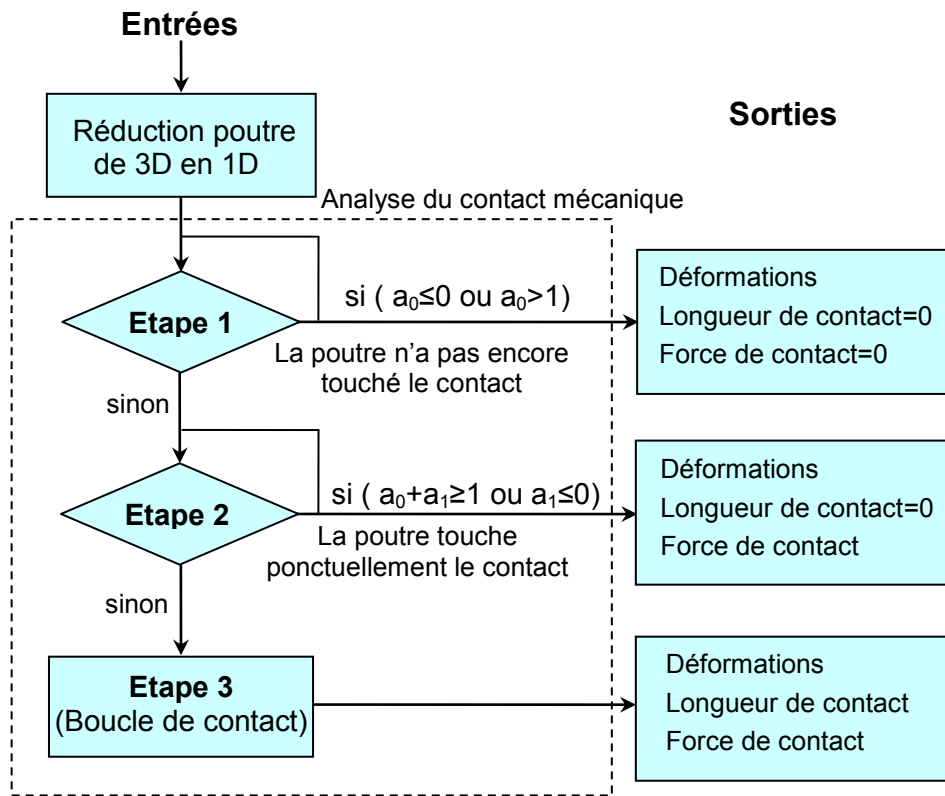


Figure II-14. Algorithme d'analyse du contact mécanique

c) *Etape 3*

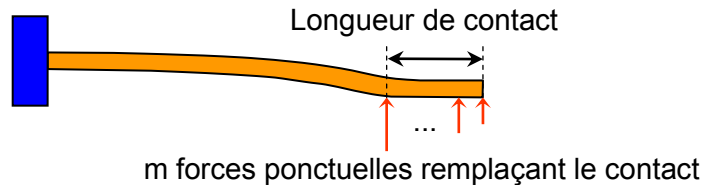


Figure II-15. Le contact est remplacé par une distribution de forces

Dans cette étape, la poutre touche le substrat avec une longueur de contact positive. Le contact est remplacé par une distribution de force comme montrée sur la Figure II-15.

Comme cette distribution de force est inconnue, la longueur de contact ne peut pas être calculée directement.

Pour trouver les forces et les couples nécessaires à la création d'une longueur de contact donnée (L_{contact}), un système de $(m + 1)$ équations de $(m + 1)$ inconnues ($a_2, a_{r-i}, i = 1 : m$) est formé :

$$\left\{ \begin{array}{l} a_0 \cdot z_{1-L_1} + a_1 \cdot z_{2-L_1} + a_2 \cdot z_{2-L_1} + \sum_{i=1}^m a_{r-i} \cdot z_{2ir-L_1} = \text{gap} \\ \dots \\ a_0 \cdot z_{1-L_m} + a_1 \cdot z_{2-L_m} + a_2 \cdot z_{2-L_m} + \sum_{i=1}^m a_{r-i} \cdot z_{2ir-L_m} = \text{gap} \\ a_0 \cdot \frac{\partial z_{1-L_m}}{\partial x} + a_1 \cdot \frac{\partial z_{2-L_m}}{\partial x} + a_2 \cdot \frac{\partial z_{2-L_m}}{\partial x} + \sum_{i=1}^m a_{r-i} \cdot \frac{\partial z_{2ir-L_m}}{\partial x} = 0 \end{array} \right. \quad \text{Eq. II-25}$$

où :

✓ $(a_0+a_1+a_2)$ est la proportion des forces et couples totaux nécessaires à la création d'une longueur de contact donnée (L_{contact}).

✓ m est le nombre de forces ponctuelles remplaçant le contact.

✓ a_{r-i} ($i=1 : m$) est la valeur de la force i , remplaçant le contact.

✓ $L_1 = L_{\text{total}} - L_{\text{contact}} \cdot \frac{m-i+1}{m}$ ($i=1 : m$) est le point d'application de la force i

remplaçant le contact, L_{total} est la longueur totale de la poutre.

✓ $z_{1-L_i}, \frac{\partial z_{1-L_i}}{\partial x}$ ($i=1 : m$) sont respectivement la déformation et la dérivée de

la déformation de la poutre en L_i , créées par toutes les forces et tous les couples, poutre de type simple encastrement (Figure II-13- (a)).

✓ $z_{2-L_i}, \frac{\partial z_{2-L_i}}{\partial x}$ ($i=1 : m$) sont respectivement la déformation et la dérivée de

la déformation de la poutre en L_i , créées par toutes les forces et tous les couples, poutre de type encastrement-posé (Figure II-13- (b)).

✓ $a_{r-i} \cdot z_{2ir-L_k}, a_{r-i} \cdot \frac{\partial z_{2ir-L_k}}{\partial x}$ ($i, k=1 : m$) sont respectivement la déformation et

la dérivée de la déformation de la poutre en L_k , créées par la force i remplaçant le contact, poutre de type encastrement-posé (Figure II-13-b).

Un algorithme de recherche dichotomique, comme montré sur la Figure II-16, est utilisé pour trouver la longueur de contact.

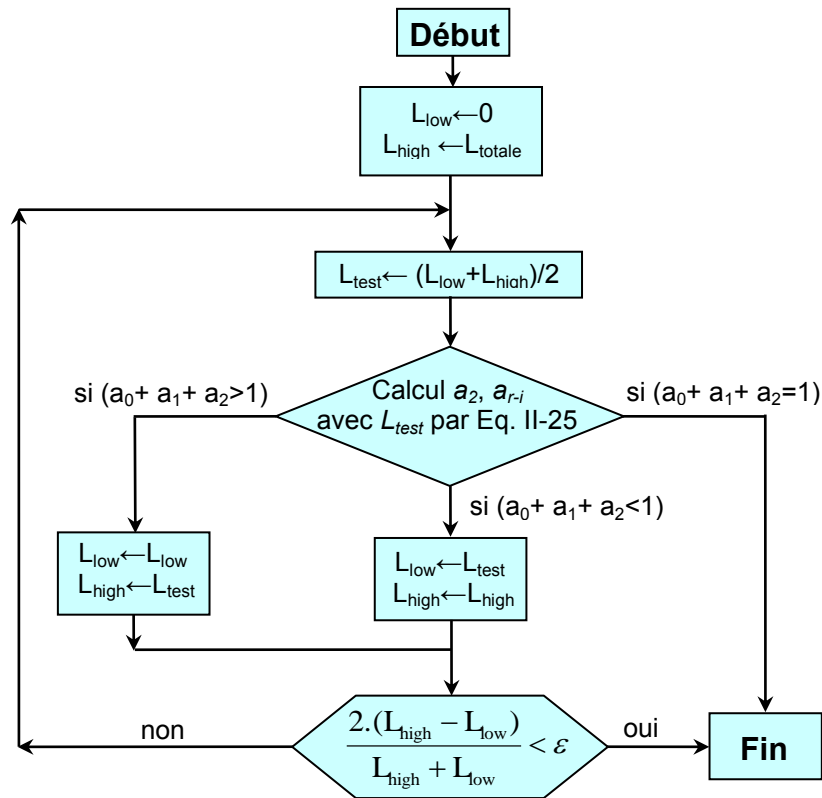


Figure II-16. Algorithme de recherche dichotomique dans l'analyse du contact mécanique

La force de contact peut être calculée par deux méthodes :

✚ Méthode directe

$$F_{\text{contact}} = F_{\text{totale}} - F_{\text{réaction-A}} \quad \text{Eq. II-26}$$

où F_{totale} est la force totale appliquée sur la poutre et $F_{\text{réaction-A}}$ est la force de réaction en A.

✚ Méthode indirecte

$$F_{\text{contact}} = F_{\text{réaction-B}} + \sum_{i=1}^m a_{r-i} \quad \text{Eq. II-27}$$

où $F_{\text{réaction-B}}$ est la force de réaction en B et $\sum_{i=1}^m a_{r-i}$ est la somme des forces remplaçant le contact.

Ces deux méthodes ont été implémentées. Elles sont équivalentes en terme de mise en œuvre et surtout en terme de résultats.

II.3.5. Validation par éléments finis

Le modèle semi analytique de calcul de déformation de la poutre avec analyse du contact mécanique a été validé par modélisation éléments finis en utilisant le logiciel Ansys® [ANS]. Nous avons défini une poutre, dont la géométrie est montrée sur la Figure II-17, et appliqué trois forces ponctuelles. Sous l'action de ces forces, la poutre se déforme et crée un contact.

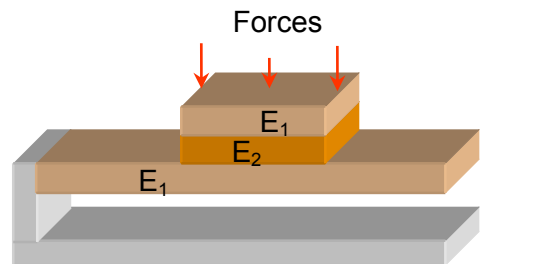


Figure II-17. Géométrie de la poutre pour valider le modèle mécanique

La comparaison, entre le modèle semi-analytique développé et le modèle éléments finis, via le tracé de la variation de la longueur du contact et de la force de contact en fonction de force appliquée sur la poutre, est donnée à la Figure II-18.

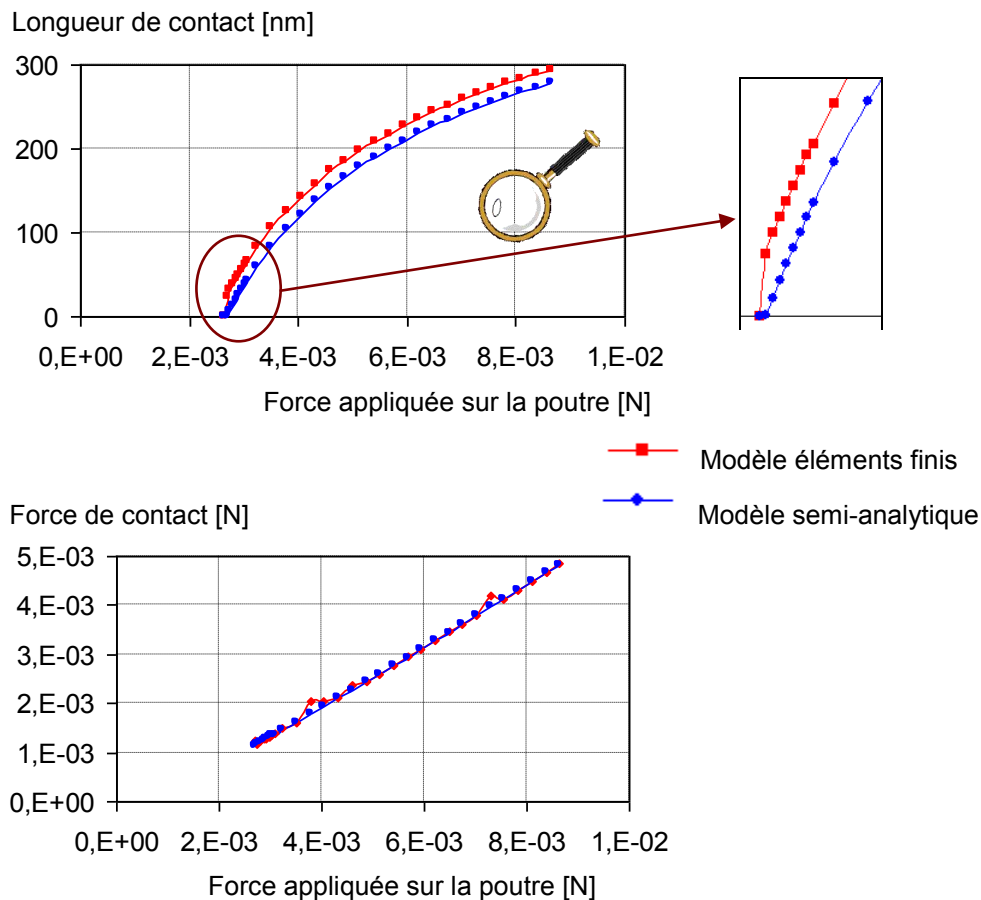


Figure II-18. Résultats de validation du modèle mécanique par éléments finis

Conclusions sur la validation

D'une manière générale, les résultats du modèle semi analytique coïncident avec ceux du modèle éléments finis. Sur la longueur du contact, il y a un petit écart entre les deux, qui vient de la discontinuité du modèle éléments finis au point de la création du contact surfacique (voir le zoom sur la longueur du contact de la Figure II-18). Au niveau des temps de calcul, on peut noter que, pour chaque point de ces courbes, le modèle éléments finis donne les résultats en environ 100000 fois plus de temps que le modèle semi analytique. Le modèle magnéto-mécanique semi-analytique ainsi que cette comparaison pour le modèle mécanique ont été présentés à CEFC 2010 [PHA-10-1] et publié dans IEEE Transaction on Magnetics [PHA-11-1]. De plus, un outil de validation éléments finis magnéto-mécanique a été entrepris par N. Galopin au laboratoire G2Elab. Nous avons présenté des premiers résultats prometteurs de ce travail à Compumag 2011 [GAL-11].

II.4. Couplage magnéto-mécanique

II.4.1. Notions de couplage multi-physiques

Une analyse de couplage multi-physiques est une combinaison d'analyses de différentes disciplines qui interagissent pour résoudre un problème global. Certaines analyses de couplage multi-physiques sont unidirectionnelles. Par exemple, dans un problème de stress-thermique, le champ de températures introduit des contraintes thermiques sur le champ structurel, mais les tensions structurelles n'affectent généralement pas la distribution de température. Ainsi, il n'est pas nécessaire d'itérer entre les deux solutions des champs. Les cas plus complexes impliquent le couplage bidirectionnel. Par exemple, notre cas de couplage magnéto mécanique, nécessite d'itérer entre les deux champs pour converger vers la solution finale.

On distingue deux méthodes de couplage par leur niveau d'échange d'informations entre les champs :

✓ Couplage fort ou couplage serré : ce couplage nécessite une seule analyse. Cette analyse implique tous les termes nécessaires du problème multi physique. Le couplage fort est avantageux lorsque l'interaction des champs couplés est fortement non linéaire. Mais, il est plus difficile à mettre en œuvre et peut poser des problèmes de convergence [ANS-05].

✓ Couplage faible ou couplage lâche : ce couplage est une boucle de résolution des champs où les résultats d'un champ sont appliqués comme source d'excitation d'un autre champ. Lorsque l'interaction des champs couplés n'est pas fortement non linéaire, le couplage faible est avantageux et flexible. De plus, il est plus facile à mettre en œuvre, car on peut modéliser des champs indépendamment.

II.4.2. Couplage magnéto-mécanique

La Figure II-19 montre le principe de couplage faible que nous avons choisi pour notre couplage magnéto-mécanique.

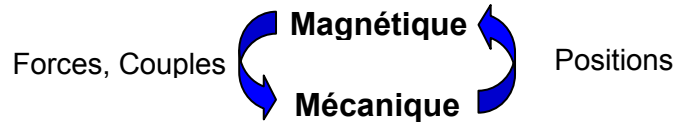


Figure II-19. Couplage faible magnéto-mécanique

L'interaction se fait via les vecteurs des forces, des couples et des positions. Le modèle magnétique nécessite les positions (la déformation) de la poutre pour calculer les forces et les couples. Ces forces et couples sont ainsi utilisés dans le modèle mécanique pour calculer les positions de la poutre.

Mathématiquement, ce couplage est équivalent à un système implicite de deux vecteurs inconnus : vecteur des positions (X) et vecteur des forces, couples (Y).

$$\begin{cases} X = f(Y) \\ Y = g(X) \end{cases} \quad \text{Eq. II-28}$$

Pour résoudre le système de couplage (Eq. II-28), la méthode de Gauss est utilisée. Pour chaque itération, le vecteur des positions (X) et l'erreur relative sont calculés par :

$$\begin{cases} X_0 = 0 \\ X_{i+1} = f(g(X_i)) \end{cases} \quad \text{Eq. II-29}$$

$$\text{erreur}_i = \text{Max}_{k=0:n} \left(\left| \frac{X_{i+1}[k] - X_i[k]}{X_{i+1}[k]} \right| \right) \quad \text{Eq. II-30}$$

Cette méthode de Gauss converge très vite pour le couplage magnéto-mécanique du nano commutateur. La Figure II-18 montre l'évolution de l'erreur relative en fonction du nombre d'itération.

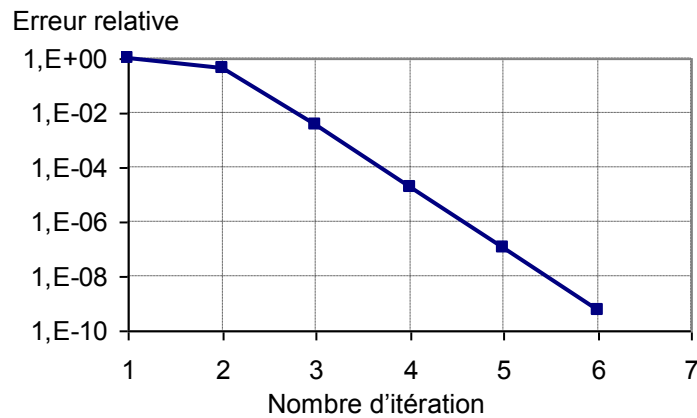


Figure II-20. Convergence de la méthode de Gauss pour couplage magnéto-mécanique du nano commutateur

Nous avons montré, dans [MOK-11], que ce couplage pouvait être amélioré en appliquant une résolution partielle de chacun des champs à chaque itération. En effet, le calcul des efforts magnétiques nécessite une intégration numérique dont la précision peut être réglée. De la même manière, le calcul du contact passe par un algorithme dont on peut ajuster la précision. Ainsi, en demandant une précision croissante en fonction de l'itération de couplage, nous parvenons à améliorer le temps global par trois.

II.5. Introduction du composant déformable dans MacMMems

II.5.1. Interface Homme / Machine (IHM)

L'IHM de MacMMems, illustrée à la Figure II-21, s'articule autour d'une fenêtre principale de type 3D. Cette fenêtre permet de visualiser et d'interagir avec un MEMS/NEMS virtuel à modéliser.

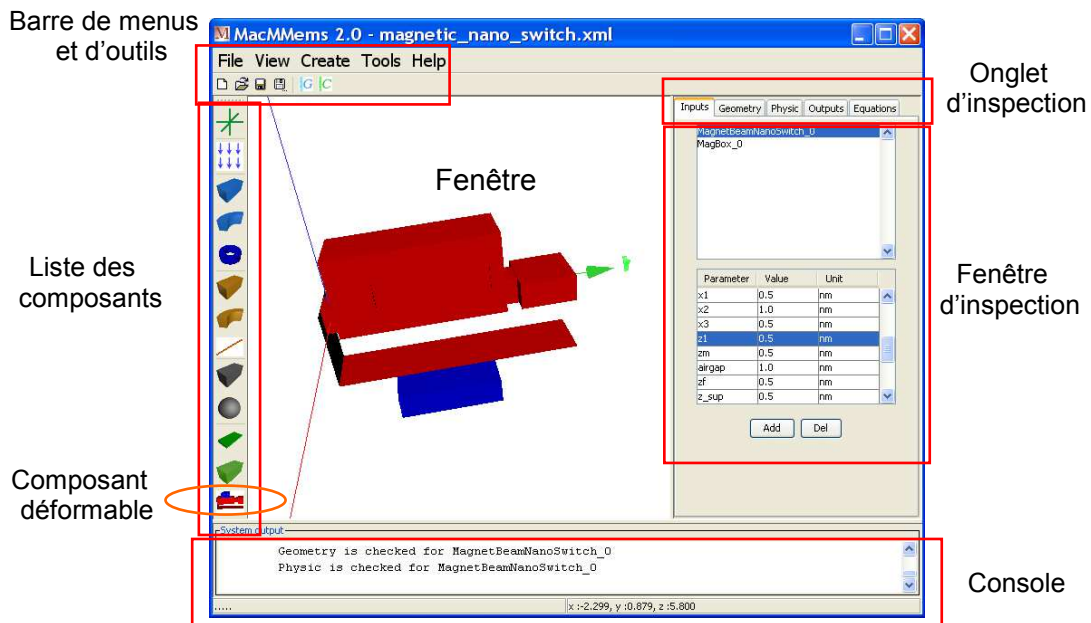


Figure II-21. IHM de MacMMems

Dans cette section, on détaillera la manipulation du composant déformable implémenté dans MacMMems. Pour plus de détails sur l'utilisation de ce logiciel MacMMems, nous recommandons au lecteur son manuel d'utilisation [PHA].

Le composant déformable est décomposé en parallélépipèdes. Les parallélépipèdes sont différenciés par leur géométrie et/ou par leurs caractéristiques physiques (magnétique, mécanique).

a) Paramétrage géométrique

Si on active l'onglet « Geometry » du composant déformable. Des paramètres apparaissent comme dans la Figure II-22-a :

- ✓ *Number Area* : est le nombre de parallélépipèdes du composant déformable.
- ✓ *Length, Width, Height* : sont respectivement des vecteurs de dimension en y, en x et en z des parallélépipèdes du composant déformable.
- ✓ *Xposition, Yposition, Zposition* : sont respectivement des vecteurs de position en x, en y et en z des parallélépipèdes par rapport du centre du premier parallélépipède.
- ✓ *Airgap* : est l'espace au dessous de la poutre (scalaire).

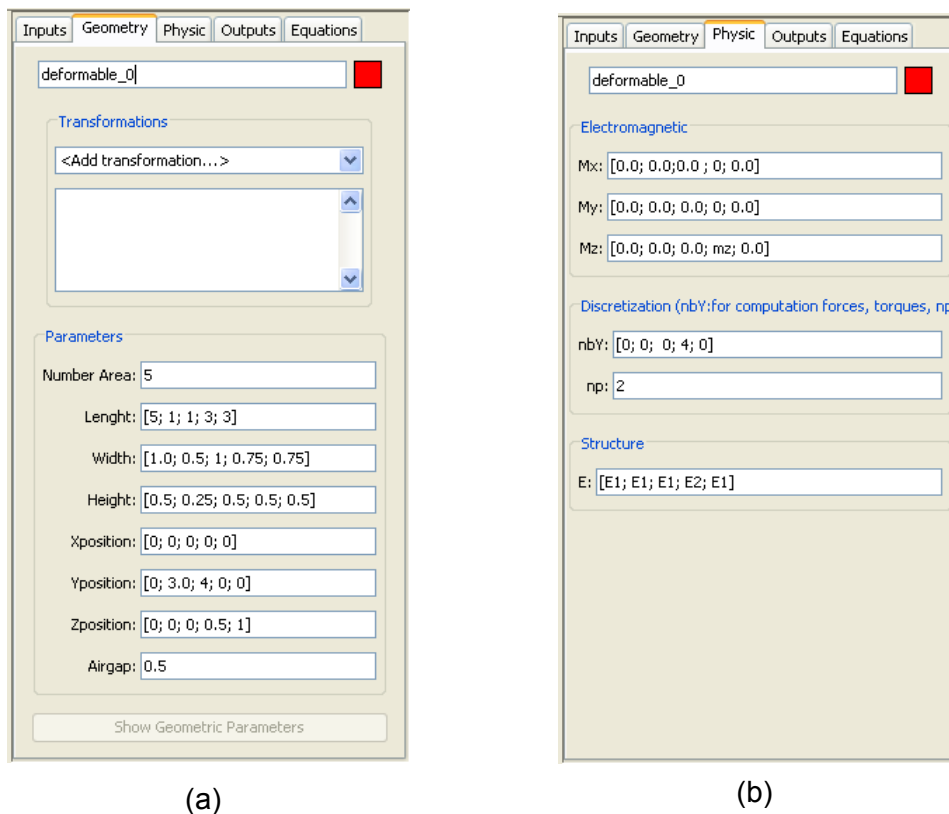


Figure II-22. (a) Paramètres géométriques et (b) Paramètres physiques du composant déformable

b) Paramétrage physique

Si on active l'onglet « Physic » du composant déformable. Des paramètres physiques apparaissent comme dans la Figure II-22-(b) :

✓ *Electromagnetic* (M_x , M_y , M_z) : sont respectivement des vecteurs d'aimantation en x , en y et en z des parallélépipédiques.

✓ *Discretization* : les deux paramètres de discrétisation sont :

- nbY : vecteur de nombre de discrétisation pour le calcul des forces et des couples sur les aimants parallélépipédiques. La Figure II-23 montre l'exemple d'un aimant parallélépipédique qui est décomposé en $nbY=4$ (4 forces et 4 couples pour le couplage).

- np : un nombre scalaire pour tracer la déformation (plus np est grand, plus il y aura des points dans les tracés, et plus la représentation sera lisse).

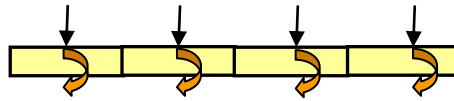


Figure II-23. Forces et couples sur un aimant parallélépipédique décomposé en $nbY=4$ pour le couplage

c) *Ajout des équations et des fonctions*

MacMMems donne aux utilisateurs la possibilité d'ajouter des équations et des fonctions en activant l'onglet « Equation » comme le montre la Figure II-24.

✓ Ajout des équations : des équations supplémentaires sont utiles pour définir les fonctions des objectifs ou des contraintes pour l'optimisation. Comme le montre l'exemple de la Figure II-24, l'équation pour calculer le volume de l'aimant est ajoutée à la première ligne dans l'onglet « Equation ».

✓ Ajout des fonctions : cette fonctionnalité a été développée pour la définition de forces supplémentaires au couplage magnéto-mécanique comme la force de Casimir. Ces fonctions sont des fonctions de densité de force (par unité de surface, voir section II.2) :

- Unité [SI] (m, N)
- Syntaxe $df_ds(variable)=...$
- 2 Exemples :

```
df_ds (d1)=10E-9/d1 ;
df_ds (dx)=1E-9+sin (dx) /dx ;
```

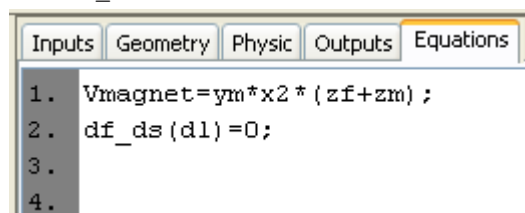


Figure II-24. Ajout des équations et des fonctions

II.5.2. Visualisation de la déformation

Après avoir généré le composant de calcul (ICAr), le *CadesCalculator* est utilisé pour étudier le comportement du modèle. Il permet aussi de tracer la caractéristique d'une grandeur de sortie en fonction d'une ou plusieurs grandeurs d'entrée.

Pour le nano commutateur magnétique, on peut tracer la déformation au long de la poutre comme le montre la Figure II-25.

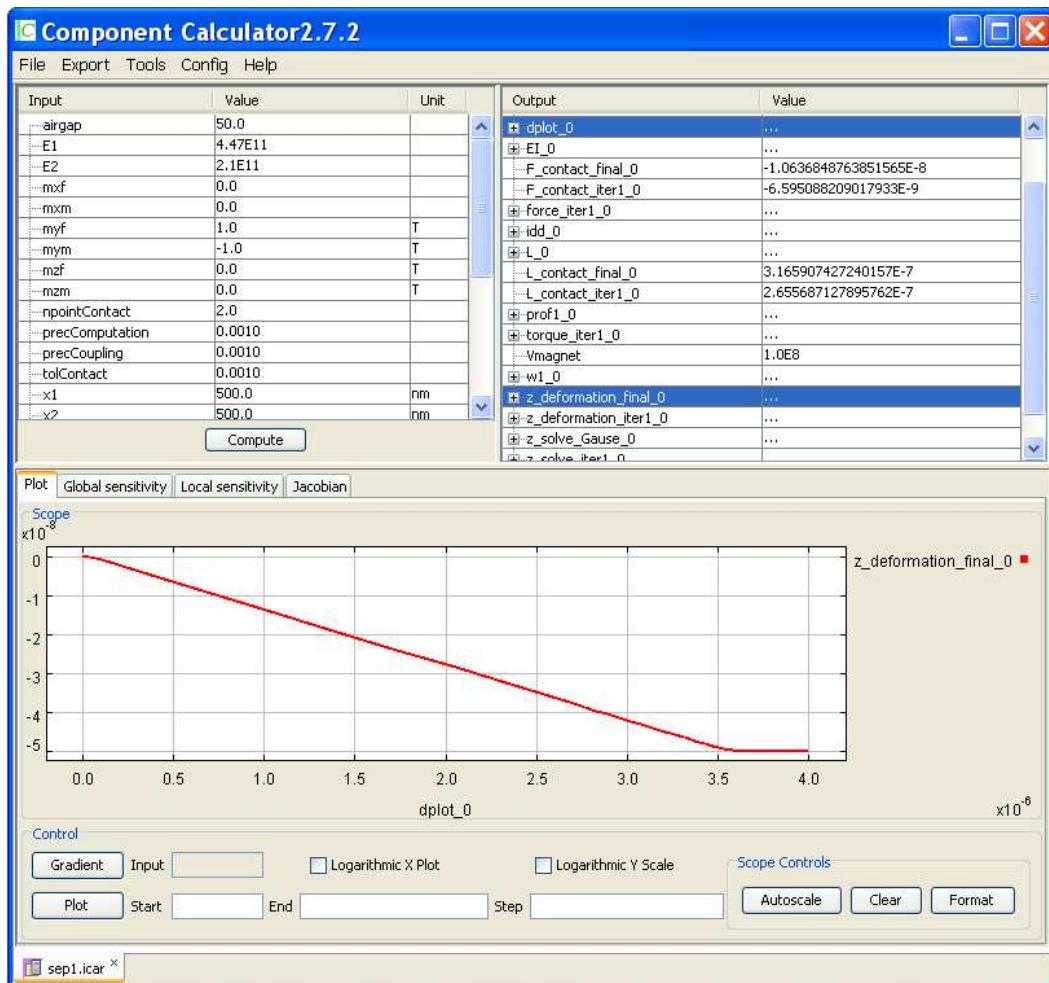


Figure II-25. Tracer de la déformation de la poutre par CadesCalculator

II.6. Conclusions

Dans ce chapitre, nous avons présenté le modèle semi analytique et outil de modélisation de couplage magnéto-mécanique dédié au nano commutateur magnétique. Pour ce couplage, le modèle magnétique est existant dans le logiciel MacMMems. L'empilement spécifique FM/AF, utilisé dans le nano commutateur, peut être modélisé comme un aimant. Le modèle semi analytique pour calculer la déformation de la poutre avec l'analyse de contact mécanique a été développé et validé par le logiciel éléments finis. Le couplage magnéto-mécanique a été implémenté dans le logiciel MacMMems avec l'algorithme de couplage choisi est Gauss-Seidel. MacMMems donne aux utilisateurs d'ajouter des équations supplémentaires comme la force de Casimir (sous forme densité surfacique de force). Ce logiciel a été livré aux autres partenaires du projet MONACO.

Chapitre III

JAP : Java Jacobian Automatic Programming

SOMMAIRE

Chapitre III

JAP : Java Jacobian Automatic Programming	65
III.1. Motivations pour JAP	67
III.1.1. Introduction	67
III.1.2. Outils existants de dérivation automatique en Java.....	67
III.1.3. Motivation pour JAP.....	69
III.2. Principes et techniques de Dérivation Automatique	69
III.2.1. Les modes de Dérivation Automatique	70
III.2.1.1. <i>Le mode direct</i>	71
III.2.1.2. <i>Le mode inverse</i>	72
III.2.2. Les stratégies d'implémentation de la DA	73
III.2.2.1. <i>Surcharge d'opérateurs</i>	73
III.2.2.2. <i>Génération de code</i>	74
III.3. Principe, architecture et limitation de JAP	76
III.3.1. Principe de DA de JAP en mode direct.....	76
III.3.1.1. <i>Objet derivative ∇u de JAP : Jdouble1</i>	76
III.3.1.2. <i>Propagation de j-méthodes</i>	77
III.3.1.3. <i>Exploitation du Jacobien et calcul de dérivées partielles</i>	79
III.3.1.4. <i>Gestion de sous fonctions externes</i>	80
III.3.2. Architecture de JAP	82
III.3.3. Limitations de JAP	84
III.4. Performances de JAP	85
III.5. Usages de JAP.....	88
III.5.1. Mise en oeuvre de JAP dans CADES.....	88
III.5.1.1. <i>Description d'un modèle de dimensionnement dans CADES</i>	88
III.5.1.2. <i>JAP_ExternalFunctionGenerator : Générateur de fonctions externes algorithmiques</i>	91
III.5.1.3. <i>JAPGenerator : Générateur de modèle de dimensionnement sml</i>	92

III.5.2. Exemples d'utilisation de JAP dans CADES	94
III.5.2.1. Validation et performances de JAP dans CADES.....	94
III.5.2.2. Modélisation et optimisation du système de suspension magnétique..	96
III.5.2.3. Utilité d'un code 100% Java	99
III.5.2.4. Optimisation du nano commutateur magnétique.....	101
III.5.3. Utilisation de JAP pour la simulation dynamique	102
III.6. Conclusions.....	104

Résumé du chapitre

Ce chapitre concerne une méthode de dérivation automatique de code en Java et son implémentation dans l'outil JAP. Tout d'abord, les limites des outils existants de dérivation automatique en Java seront présentées. Ensuite, nous présenterons le principe et les techniques de dérivation automatique pour bien comprendre les détails de l'architecture de JAP. Le test de performance de JAP sera aussi présenté en comparant avec l'outil existant ADOL-C. Nous finirons enfin par les usages de JAP pour les outils de dimensionnement au G2Elab avec des exemples d'application ainsi qu'un moyen pour orienter des systèmes d'équations différentielles algébriques (DAE) vers des équations différentielles ordinaires (ODE).

III.1. Motivations pour JAP

III.1.1. Introduction

La connaissance des dérivées est utile pour plusieurs applications comme l'identification des paramètres, l'analyse de sensibilité, l'analyse de l'incertitude, l'optimisation etc. Pour ces applications, l'évaluation précise des dérivées est souvent exigée. Les techniques de calcul des dérivées sont présentées dans la section I.3.4.1. du premier chapitre. Parmi ces techniques, celle de Dérivation Automatique (DA) fournit un moyen efficace pour évaluer précisément des gradients des fonctions représentées par des programmes.

La DA transforme le programme, représentant la fonction mathématique, en une autre fonction qui fait le calcul de la fonction aussi que celui de ses dérivées. En combinant les dérivées des opérations élémentaires, dont les dérivées sont déjà connues, selon la règle des fonctions composées du calcul différentiel, un outil de DA est capable de différencier les fonctions de complexité arbitraire.

Profitant des avantages de la DA, différents outils ont été mis en œuvre jusqu'à présent. De cette façon, la DA a été mise à la disposition de plusieurs langages de programmation. Une base de données sur les outils ainsi que sur les publications concernant la DA, est régulièrement mise à jour sur <http://www.autodiff.org/>. La plupart de ces outils sont gratuits et/ou utilisables en ligne (par exemple TAPENADE [HAS-04], [TAP]).

III.1.2. Outils existants de dérivation automatique en Java

Sur la base de données actualisée régulièrement [AUT], on peut trouver un certain nombre d'outils de DA pour les langages de programmation comme Fortran, C/C++ ou Matlab. En ce qui concerne le langage portable Java, il est pratiquement inexistant dans le monde de la DA. Cependant, il existe quelques tentatives et/ou outils :

✚ Dans les travaux de Master en 2001 de Rune Skjelvik [SKJ-01], les auteurs ont proposé un outil implémentant une simulation de surcharge d'opérateurs (voir la section III.2). La limitation majeure de cet outil est qu'il ne supporte que certaines instructions du langage Java et que les fonctions travaillant sur des scalaires (les

données de type tableau ([]) ne sont pas supportées). Néanmoins, ces travaux ont montré la faisabilité d'un outil de dérivation automatique en Java.

✚ *FGOT* [COU] : Cet outil a été développé à la suite de la thèse de M. Caldora Costa [CAL-01] au G2Elab. A partir de 2002, *FGOT* utilise l'interpréteur Jython [JYT] pour analyser les expressions algébriques. Jython étant lui-même écrit en Java, la surcharge des opérateurs usuels (+, -, *, /, **) et des fonctions intrinsèques (sin, cos, ...) devient possible, ce que ne permet pas le langage Java seul. Il est également possible d'ajouter ses propres fonctions. Il est à noter que les surcharges ainsi définies dans *FGOT* gèrent non seulement les dérivées, mais aussi l'analyse des intervalles [CAP-02]. Cependant, ce logiciel ne dérive pas d'algorithmes et ne propose pas de dérivation sélective.

✚ *JavaDiff* : cet outil a été développé au G2Elab en 2004 pendant la thèse de V. Fischer [FIS-04], [FIS-05]. *JavaDiff* est similaire à l'outil [SKJ-01] qui remplace tous les méthodes mathématiques par des méthodes Java. Pendant le DEA de E. Dezille au G2Elab [DEZ-04], l'auteur a utilisé *JavaDiff* pour dériver des modèles de dimensionnement de CADES appliqué pour l'optimisation d'un actionneur linéaire basé sur un modèle en réseau de réluctance. Toutefois, ces utilisations n'étaient pas complètement automatiques et n'étaient applicables qu'à des modèles scalaires.

✚ *ADiJaC* : ce dérivateur a été développé pendant la thèse de E.I. Slusanschi [SLU-08], dirigé par le Professeur Christian H. Bischof, à l'« *Institute for Scientific Computing of RWTH Aachen University* ». *ADiJaC* génère le code de calcul dérivatif en analysant le byte code compilé. Nous avons travaillé en commun avec E.I. Slusanschi [PHA-11-2] pour évaluer leur outil, mais nous n'avons pas pu le mettre en œuvre nous même. Cet outil de DA en Java montre les limites suivantes :

✓ Il peut dériver des fonctions en présence des tableaux si on intervient manuellement dans le code généré pour initialiser ces tableaux. L'automatisation de cette étape par les auteurs d'*ADiJaC* est en cours, cette limite devrait être levée.

✓ Il ne peut pas dériver des fonctions en présence de sous fonctions externes. Cette propriété est très importante pour dériver des modèles complexes utilisant des bibliothèques de calcul exploitant d'autres techniques de calcul de dérivées comme le théorème des fonctions implicites (voir la section III.3.1.4.).

✓ La sélectivité n'est pas disponible dans *ADiJaC*. Cette fonctionnalité de dérivation sélective est primordiale pour réduire le temps de calcul et aussi pour

réduire la mémoire utilisées pendant le processus d'évaluation des dérivées (voir les détails de la sélectivité dans la section III.3.1.3.).

III.1.3. Motivation pour JAP

Compte tenu des limites des outils existants de DA en Java présentés ci-dessus et de la demande de notre équipe de recherche au G2Elab, nous avons développé un tel outil, générique, qui a été nommé JAP pour Java Jacobian Automatic Programming.

JAP doit surmonter les limites des outils actuels de DA en Java en répondant en particulier à nos besoins :

- ✓ Être automatique et facile à utiliser.
- ✓ Supporter toutes les fonctions complexes écrites en langage Java avec des données de type tableau en une dimension ([]) et en deux dimensions ([[]]).
- ✓ Supporter les sous fonctions externes fournies sous forme de bibliothèques de calcul (archive JAR -.jar).
- ✓ Supporter un calcul sélectif du Jacobien selon les besoins d'exploitation.

Avec ce cahier des charges, l'outil JAP a été développé et implémenté dans les outils de dimensionnement au G2Elab (voir la section III.5).

Il existe toutefois quelques limites que nous n'avons pas cherché à dépasser, en particulier, celle de la dérivation d'un ensemble de classes Java incluant tout particulièrement les notions objets telles que l'héritage.

III.2. Principes et techniques de Dérivation Automatique

Pour bien comprendre la démarche de mise en œuvre utilisée dans l'outil JAP, un bref aperçu des différents modes et techniques de DA sera présenté.

La DA évalue précisément des dérivées avec un effort minimal de préparation du code à dériver. Toutefois, il faut être conscient que la DA ne résout pas les limites mathématiques de dérivation bien connues en certains points non continus et/ou non dérivables des fonctions mathématiques. C'est par exemple la discontinuité et la non dérivabilité créées par les instructions conditionnelles telles que *if...else...* La Figure III-

1 montre un exemple d'une fonction non continue et non dérivable au point x_0 , la DA considère que $\frac{\partial y}{\partial x}\Big|_{x_0} = 0$.

```
if (x < x0) {
    y = y0 ;
} else {
    y = y1 ;
}
```

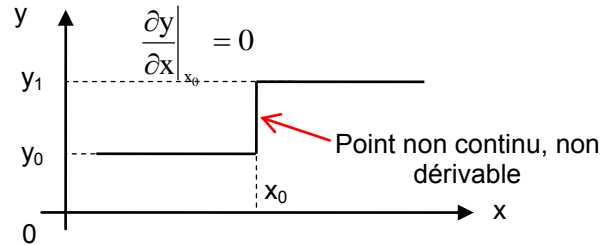


Figure III-1. Exemple de non dérivabilité de la fonction

La DA peut être considérée comme une transformation sémantique appliquée à un certain code de calcul. Chaque calcul, exprimé sous la forme d'un programme informatique, peut être considéré comme une longue suite d'opérations élémentaires comme l'addition, la soustraction, la multiplication ou la division, ainsi que des fonctions intrinsèques (exp, sin, cos, etc.). Ainsi, le concept clé de la DA est d'appliquer la règle de composition de ces opérations élémentaires, et de ces fonctions intrinsèques dont les dérivées sont déjà connues. Ces dérivées sont propagées à chaque étape de l'arbre de calcul de façon à obtenir, par composition, les dérivées de l'ensemble du programme. Les différentes techniques de propagation mènent toutes aux mêmes Jacobiens, mais ont des coûts différents en terme de temps de calcul, et de mémoire utilisée. Deux techniques bien documentées sont le mode direct et le mode inverse. Une brève introduction à ces deux modes est présentée dans les paragraphes suivants.

III.2.1. Les modes de Dérivation Automatique

Considérons une fonction f représentée par un programme :

$$f : \mathfrak{R}^n \mapsto \mathfrak{R}^m \quad \text{Eq. III-1}$$

avec n variables d'entrée x_i ($i \in [1 ; n]$), on cherche à évaluer la matrice jacobienne, de dimension $m \times n$, des $\partial f_j / \partial x_i$ ($j \in [1 ; m]$).

La fonction est calculée par une longue suite d'opérations élémentaires représentées par le graphe de calcul comme illustré dans la Figure III-2. Il y a deux modes de DA suivant le sens de propagation : propagation directe et propagation inverse.

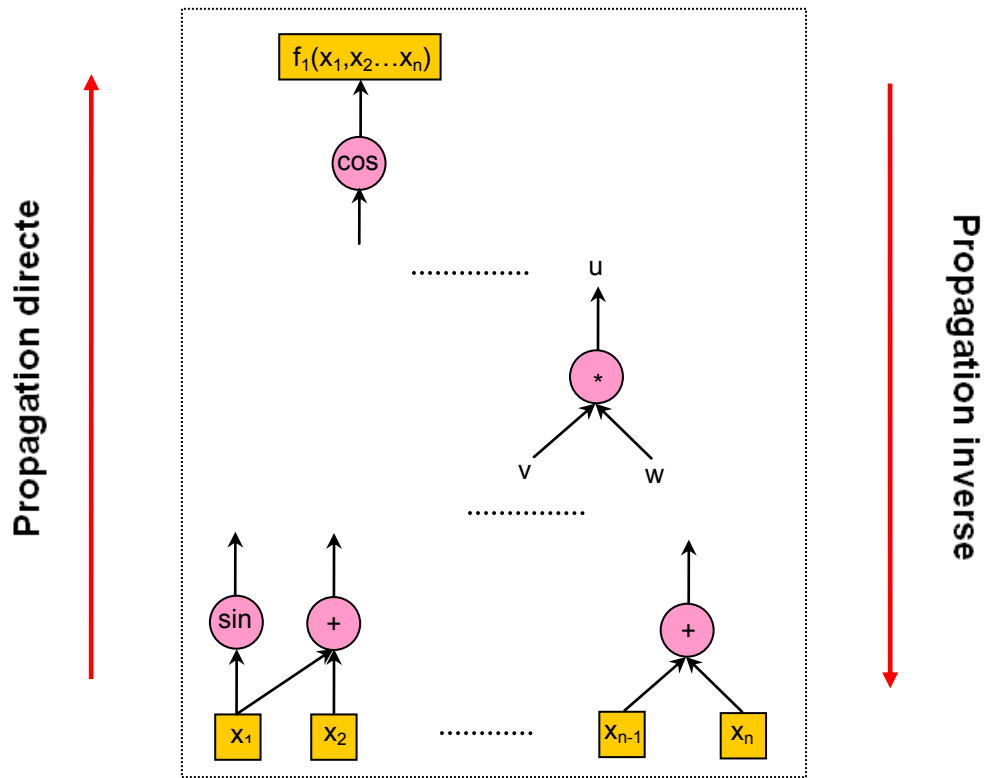


Figure III-2. Calcul de la fonction représentée par le graphe de calcul

III.2.1.1. Le mode direct

Comme son nom l'indique, le mode direct propage les valeurs des dérivées dans le même sens que le calcul de la fonction f (voir la Figure III-2). Dans ce mode, un objet « *derivative* » ∇u de type tableau de taille n est associé à chaque variable scalaire dans le programme de telle sorte que cet objet contienne des dérivées de u par rapport à n variables d'entrée. Les valeurs de l'objet *derivative* ∇u sont changées quand la variable u elle-même est changée.

Par exemple, pour le code qui contient l'instruction :

$$u(x) = v(x) * w(x) \tag{Eq. III-2}$$

avec $x \in \mathbb{R}^n$, l'objet dérivative ∇u se calcule par :

$$\nabla u = v \cdot \nabla w + w \cdot \nabla v \tag{Eq. III-3}$$

où $\nabla v, \nabla w$ sont respectivement des objets *derivatives* qui contiennent les dérivées de v et w par rapport aux n variables d'entrée x .

De cette façon, l'information des dérivées est propagée avec l'évaluation de f . Au début, ∇x est initialisé comme un vecteur unitaire $1 \times n$. A la fin on obtient les dérivées désirées par rapport aux variables d'entrée x : $\partial f / \partial x = \nabla f$.

Dans le code dérivatif ainsi obtenu, les objets *derivatives* sont des tableaux de dimension n et le calcul de leurs composantes respectives introduit des boucles supplémentaires pouvant pénaliser les temps de calcul. Pour améliorer les performances du calcul de la matrice jacobienne complète ($m \times n$), il est possible de paralléliser ces boucles supplémentaires [BUC-04], si n (la taille de ∇u) est assez grand. Le coût d'évaluation des dérivées (mémoires utilisées et temps de calcul), en termes de nombre d'opérations, augmente linéairement avec la taille de l'objet *derivative* ∇u .

Si on veut calculer une partie de la matrice jacobienne ($m \times d$), $1 \leq d \leq n$: les dérivées de f par rapport à d paramètres d'entrées, l'objet *derivative* ∇u est de taille d . Les mémoires utilisées et le temps de calcul sont proportionnels à d en comparaison avec l'évaluation de la fonction originale f . Ce mode de propagation directe montre un avantage majeur de sélectivité si $d \ll n$.

III.2.1.2. Le mode inverse

Contrairement au mode direct, le mode inverse propage les dérivées dans le sens inverse du calcul de la fonction f (voir la Figure III-2). Dans ce mode, un objet adjoint \bar{u} de type tableau de taille m est associé avec chaque variable scalaire intermédiaire u . L'objet adjoint \bar{u} contient les valeurs dérivées de la fonction f par rapport aux variables intermédiaires u .

Les informations adjointes se propagent dans l'ordre inverse de l'exécution du programme d'origine. On part de l'objet connu $\bar{f} = \partial f / \partial f$ formé d'un vecteur unitaire, pour retourner à l'objet désiré $\partial f / \partial x = \bar{x}$.

Pour chaque opération :

$$u = v \oplus w \quad \text{Eq. III-4}$$

où \oplus est une opération élémentaire ou fonction intrinsèque.

La propagation inverse est réalisée par l'ajout aux objets adjoints \bar{v} et \bar{w} (obtenus par les autres branches du graphe dans lesquels ces variables interviennent) des contributions respectives de $\bar{u} \cdot \partial u / \partial v$ et $\bar{u} \cdot \partial u / \partial w$:

$$\begin{aligned} \bar{v} &\leftarrow \bar{v} + \bar{u} \cdot \partial u / \partial v \\ \bar{w} &\leftarrow \bar{w} + \bar{u} \cdot \partial u / \partial w \end{aligned} \quad \text{Eq. III-5}$$

Considérons le même exemple qu'en mode direct :

$$u(x) = v(x) * w(x) \quad \text{Eq. III-6}$$

avec $x \in \mathfrak{R}^n$, les objets adjoints \bar{v} et \bar{w} se calculent par l'équation suivante, sachant que les grandeurs v, w ont été évaluées auparavant :

$$\begin{aligned}\bar{v} &\leftarrow \bar{v} + w.\bar{u} \\ \bar{w} &\leftarrow \bar{w} + v.\bar{u}\end{aligned}\tag{Eq. III-7}$$

Le mode inverse exige le calcul préalable de la fonction. Tous les résultats intermédiaires sont enregistrés ou re-calculés pour être utilisés dans la procédure du calcul adjoint.

Comme en mode direct, le coût d'évaluation des dérivées (mémoires utilisées et temps de calcul), en termes de nombre d'opérations, augmente en fonction de la taille (m) de l'objet adjoint \bar{u} . Ainsi, le mode inverse de la DA est meilleur en terme de temps de calcul si $m \ll n$. En particulier, la matrice jacobienne d'une fonction scalaire ($m=1$) peut être calculée avec un petit multiple du temps d'évaluation nécessaire de la fonction d'origine, indépendamment du nombre de variables n .

III.2.2. Les stratégies d'implémentation de la DA

Dans cette section, deux stratégies principales d'implémentation, qui sont à la base de la création des outils de DA, sont présentées : La surcharge d'opérateur et la génération de code.

III.2.2.1. Surcharge d'opérateurs

La surcharge d'opérateurs est disponible dans certains langages de programmation comme C/C++ ou Fortran90. Cette technique permet de définir un nouveau sens pour les opérateurs mathématiques (+, -, *, /, etc.) pour de nouveaux types de données (nombre complexes, vecteurs, etc.)

En effet, les outils de DA s'appuyant sur la surcharge d'opérateurs redéfinissent toutes les fonctions mathématiques de base en supposant que les opérandes sont d'un type de donnée spécifique. Ces opérandes sont des objets définis par les outils de DA (l'objet *derivative* ∇u en mode direct ou l'objet adjoint \bar{u} en mode inverse). La Figure III-3 montre un exemple de définition de la classe d'un objet *derivative* ∇u en C++ avec la redéfinition (surcharge) de l'opérateur de multiplication. On peut observer que les valeurs de la fonction et des dérivées sont propagées simultanément.

```

1  class ADType {
2      protected :
3          Type val ;
4          Type dval;
5      public :
6          ADtype operator* (ADtype &arg) {
7              ADtype result;
8              result.val=val*arg.val;           //calcul valeur
9              result.dval=dval*arg.val+val*arg.dval; //calcul dérivée
10             return result;
11         }
12     }

```

Figure III-3. Exemple de la classe de définition d'un objet *derivative* ∇u en mode direct avec redéfinition (surcharge) de l'opérateur de multiplication

Grâce à la simplicité de mise en oeuvre, les outils de DA basés sur la surcharge d'opérateurs offrent aux utilisateurs la possibilité d'évaluer non seulement des dérivées de premier ordre mais aussi des dérivées d'ordre supérieur.

Nous pouvons citer des outils de DA s'appuyant sur la technique de surcharge d'opérateurs comme : ADOL-C [WAL], FADBAD [BEN-96] pour des programmes C/C++; ADOL-F [SHI-96] pour des programmes FORTRAN95; ADMIT/ADMAT [COL-00] et MAD [FOR-06] pour des programmes Matlab.

III.2.2.2. Génération de code

Cette technique d'implémentation est connue sous le nom anglais *source-to-source transformation*. Comme le nom l'indique, un outil implémentant cette technique, prend en entrée un programme initial et génère un deuxième programme augmenté, capable d'évaluer la fonction et ses dérivées. Pour générer automatiquement ce code, l'outil basé sur cette technique joue le même rôle qu'un compilateur informatique. Selon [KOW-07] la procédure de génération de code se compose de six étapes avec deux principales que sont l'analyse du code source et la génération du code cible. La Figure III-4 montre une représentation générale de l'architecture d'un outil de DA basée sur cette technique.

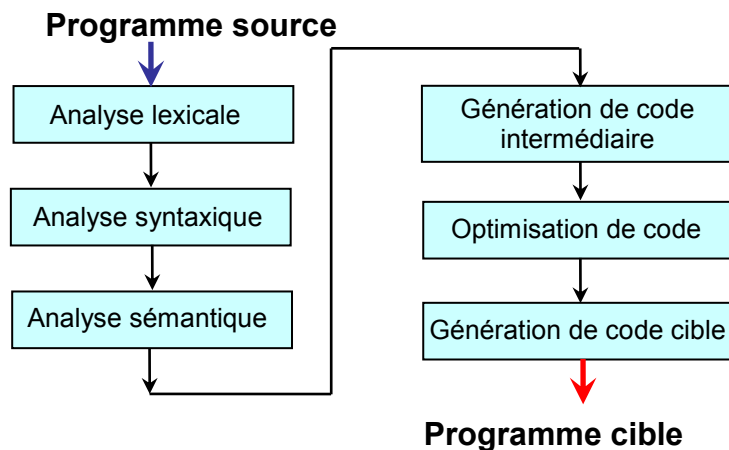


Figure III-4. L'architecture d'un outil de DA basée sur la technique de génération de code

L'avantage des outils implémentant la technique de génération de code est la capacité d'analyser, et d'optimiser au maximum le programme de calcul des dérivées. Ces optimisations sont réalisées par l'analyse d'activité des variables ou la gestion des matrices creuses par exemple. L'analyse d'activité des variables permet d'identifier des variables intermédiaires qui n'interviennent pas dans le calcul de dérivées des sorties. La dérivation de ces variables est donc ignorée.

La plupart des outils de DA supportent le langage FORTRAN comme TAPENADE [HAS-04] [TAP], ADIFOR [BIS-96], TAF [GIE-05], parce qu'il offre aux programmeurs une structure d'instructions opératives standards (affectation, branches conditionnelles, boucles, ...), et que de nombreux codes de calculs ont été développés dans ce langage. Pour les mêmes raisons, le langage C, est également bien représenté, on peut citer ADIC [BIS-97], TAPENADE [HAS-04] [TAP]. Pour les langages objets comme C++ et Java, la difficulté est plus grande. Pour le langage Java, comme il ne permet pas de surcharger les opérateurs, JAP ainsi que la plupart des outils cités dans la section III.1.2, sont basés sur cette technique de génération de code.

Il est à noter que la structure mise en œuvre dans TAPENADE permet de dériver un code décrit en C ou en Fortran, via un langage intermédiaire neutre de type langage impératif (comme C ou Fortran). Il est alors possible de bénéficier de cet outil de dérivation par la création d'un parseur et d'un générateur (*printer*) selon la Figure III-5.

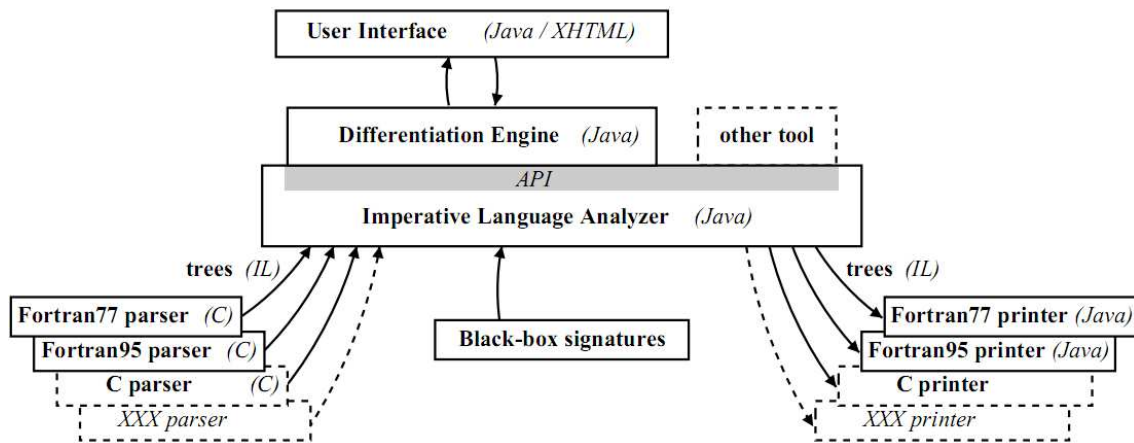


Figure III-5. L'architecture de TAPENADE (extrait de [HAS-04])

III.3. Principe, architecture et limitation de JAP

III.3.1. Principe de DA de JAP en mode direct

Outre la technique de génération de code imposée par le langage Java (ce langage ne supporte pas la surcharges des opérateurs), JAP effectue une simulation de surcharges d'opérateurs en remplaçant des opérations basiques et des fonctions intrinsèques par des méthodes Java.

III.3.1.1. Objet derivative ∇_u de JAP : *Jdouble1*

Pour propager les dérivées en mode direct, JAP transforme toutes les variables actives de type *double* en l'objet *derivative Jdouble1* (1 désigne l'ordre de dérivation). La Figure III-6 montre la définition de la classe *Jdouble1*. Cette classe possède trois variables d'instances (*double val*, *int nGrad*, *double[] dval*) pour définir les caractéristiques de l'objet *derivative Jdouble1*. Par rapport à l'objet *derivative ∇_u* standard définit dans la Figure III-3 (*ADType* propage des dérivées en mode scalaire), le *Jdouble1* est plus générique et propage des dérivées en mode vectoriel. La différence est la variable d'instance *nGrad* (*int*) qui est le nombre de variable d'entrées à dériver (dimension du champ *dval* (*double[]*)). La valeur de *nGrad* est définie par l'utilisateur en fonction de ses besoins.

```

1 public class Jdouble1 {
2     public double val;
3     public int nGrad;           // number of variable to derive
4     public double[] dval;
5
6     // constructor
7     public Jdouble1(int nGrad) {
8         this.val = 0;
9         this.nGrad = nGrad;
10        this.dval = new double[nGrad];
11    }
12
13    // simulation of method res=A*B
14    public static Jdouble1 times(int nGrad, Jdouble1 A, Jdouble1 B) {
15        Jdouble1 res = new Jdouble1(nGrad);
16        res.val = A.val * B.val;
17        for (int i = 0; i < nGrad; i++) {
18            res.dval[i] = A.dval[i] * B.val + A.val * B.dval[i];
19        }
20        return res;
21    }
22    ...
23 }

```

Figure III-6. Jdouble1 : classe de l'objet *derivative* ∇u (1^{er} ordre) de JAP

III.3.1.2. Propagation de j-méthodes

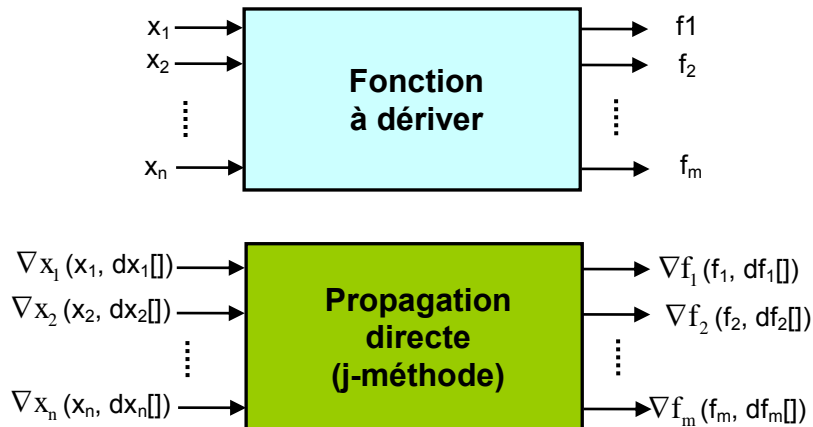


Figure III-7. Fonction à dériver et la propagation directe de j-méthodes

Pour montrer le principe de la propagation directe de j-méthode, la Figure III-8 montre l'exemple d'un algorithme de calcul à dériver fourni par un utilisateur de JAP. L'exemple choisi ici retourne un scalaire avec des arguments scalaires (a_1, k) et aussi vecteur (av). Des opérateurs standards, des fonctions intrinsèques (cos, log, abs), une boucle (for) et une branche conditionnelle (if) sont utilisés.


```

1      public double func(double a1, int k, double[] av) {
2          double res;
3          int i;
4          res = Math.abs(a1/ k);
5          for (i = 0; i < av.length; i++) {
6              res += Math.cos(Math.log(av[i]) + i);
7          }
8          if (av[0] > a1) {
9              res *= av[0]- a1;
10         } else {
11             res *= a1 / 5;
12         }
13         return res*2;
14     }

```

Figure III-8. Exemple d'une méthode de calcul à dériver

La Figure III-9 montre le code généré par JAP après analyse du programme fourni. Une méthode travaillant avec *Jdouble1* est générée. Un préfixe « j » est ajouté au nom de cette méthode (l'exemple est *jfunc*). On l'appelle *j-méthode* ou bien *j-fonction*. Pour générer la *j-méthode*, toutes les variables actives (*double*) sont transformées en *Jdouble1*, et toutes les opérations et fonctions intrinsèques sont transformées en des méthodes statiques de la classe *Jdouble1*. On remarque par exemple les appels aux fonctions *Jdouble1.div* et *Jdouble1.times* ainsi que *Jdouble1.cos* ou *Jdouble1.abs* qui correspondent en quelque sorte à la surcharge des opérateurs '/' et '*', ainsi que des fonctions intrinsèques *cos* et *abs*.

```

1      public Jdouble1 jfunc(Jdouble1 a1, int k, Jdouble1[] av) {
2          int jap_nGrad = a1.nGrad;
3          //declaration des variables locales
4          Jdouble1 res;
5          int i;
6          //CoreComputation
7          res=Jdouble1.abs(jap_nGrad,Jdouble1.div(jap_nGrad, (a1), (k)));
8          for (i = 0; i < av.length; i++) {
9              res.plusEquals(Jdouble1.cos(jap_nGrad,Jdouble1.plus(
10                 jap_nGrad, (Jdouble1.log(jap_nGrad,av[i])), (i))));
11          }
12          if (av[0].val > a1.val) {
13              res.timesEquals(Jdouble1.minus(jap_nGrad, (av[0]), (a1)));
14          } else {
15              res.timesEquals(Jdouble1.div(jap_nGrad, (a1), (5.0)));
16          }
17          return Jdouble1.times(jap_nGrad, (2.0), (res));
18     }

```

Figure III-9. Exemple de simulation de surcharge d'opérateurs de JAP en mode direct

III.3.1.3. Exploitation du Jacobien et calcul de dérivées partielles

Après avoir créé des *j-méthodes* qui propagent en même temps la fonction et aussi des dérivées par j'objet *derivative Jdouble1*, JAP crée aussi des méthodes d'exploitation de toute la matrice jacobienne ou des dérivées sélectives.

✚ Exploitation de toute la matrice jacobienne : les dérivées de toutes les sorties par rapport à toutes les entrées. Dans ce cas, *nGrad* est égal au nombre des entrées.

✚ Exploitation sélective du Jacobien : deux possibilités

- Les dérivées de toutes les sorties sur une entrée choisie par *index*.

Dans ce cas *nGrad* est égal à 1.

- Les dérivées de toutes les sorties sur quelques entrées choisies par *index[]*. Dans ce cas *nGrad = index.length*.

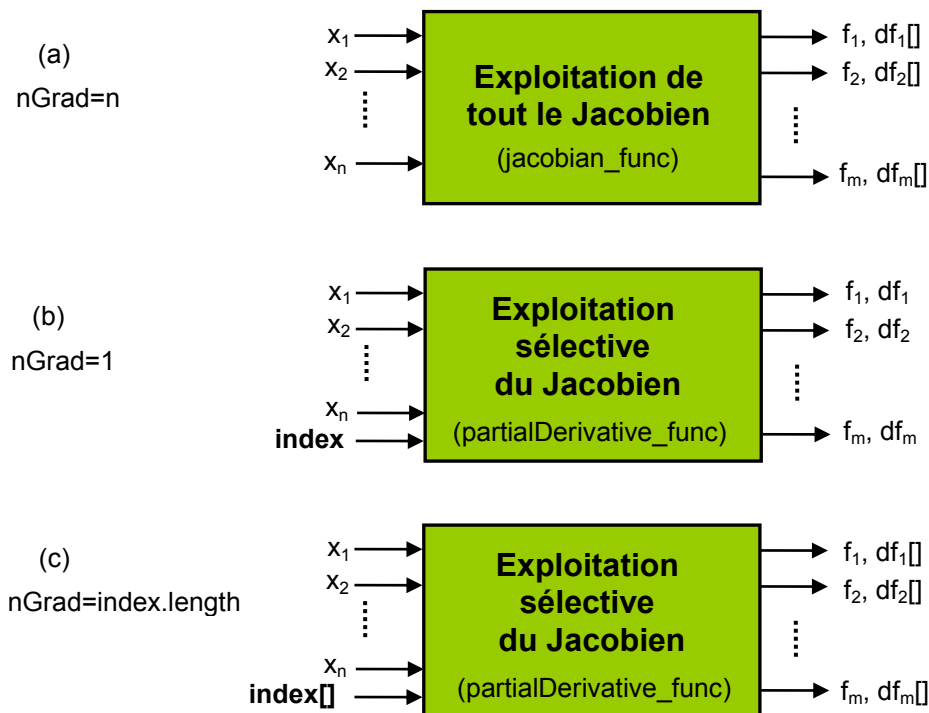


Figure III-10. Trois fonctions d'exploitation générées par JAP

```
11 //exploitation de tout le Jacobian
12     public double[] jacobian_func(double a1,int k,double[] av) {
13         ...
14     }
20 //exploitation de la dérivée par rapport à un paramètre
21     public double[] partialDerivative_func(double a1,int k,double[]av,
22     int index) {...
23     }
//exploitation des dérivées par rapport a quelques paramètres
30     public double[] partialDerivative_func(double a1,int k,double[]av,
31     int[] index) {...
32     }
```

Figure III-11. Convention de la signature des méthodes d’exploitation générées

La Figure III-11 montre un exemple standard de la signature des méthodes d’exploitation générées par JAP. Pour chaque méthode d’exploitation, elle effectue les tâches suivantes :

- ✓ Déterminer *nGrad* : le nombre de variable d’entrées à dériver qui est aussi la dimension du champ *dval* de *Jdouble1*.
- ✓ Créer et initialiser les variables *Jdouble1* correspondant à des variables actives d’entrée.
- ✓ Faire appel à la *j-méthode*.
- ✓ Exploiter les résultats.

III.3.1.4. Gestion de sous fonctions externes

JAP est aussi capable de supporter des sous fonctions externes fournies sous forme de bibliothèque de calcul (archive JAR *.jar*). Une telle bibliothèque de calcul doit fournir le calcul du Jacobien des fonctions qu’elle contient. Ce calcul peut être réalisé par n’importe quelle technique : symbolique, différences finies ou Dérivation Automatique de code (par JAP ou par d’autres outils). Par convention, le Jacobien est fourni sous forme d’une méthode de nom avec le préfixe *jacobian_*. La Figure III-12 montre un exemple standard de la signature de la sous fonction externe et de son Jacobien correspondant.

```

11 //codage de la fonction externe
12 public double extfunc(double a1,double[] av) {
    ...
20 }
21 //codage du jacobien de la fonction externe
22 public double[] jacobian_extfunc(double a1,double[] av) {
    ...
30 }
    
```

Figure III-12. Convention de la signature d'une fonction externe et de son Jacobien

Si la sous fonction externe est générée par JAP, elle fournit non seulement le Jacobien, mais aussi la *j-méthode* correspondante. Cette *j-méthode* est utilisée directement pour la composition avec le modèle global afin d'améliorer les performances.

Si la sous fonction externe est générée par une autre technique que JAP qui ne fournit que le Jacobien ou un Jacobien partiel, JAP doit faire lui-même la composition pour générer la *j-méthode* correspondante. Cette composition, et en particulier le passage entre ces deux modes de propagation de sensibilité, doit être gérée convenablement pour améliorer les performances de calcul [DEL-04].

$$dS_j = \sum_i \frac{\partial S_j}{\partial E_i} . dE_i \quad \text{Eq. III-8}$$

où S_j et E_i sont respectivement les sorties et les entrées de la sous fonction externe, la matrice des $\frac{\partial S_j}{\partial E_i}$ est le Jacobien fournit et dS_j , dE_i sont respectivement des *Jdouble1* de sorties et d'entrées de la *j-méthode* à générer.

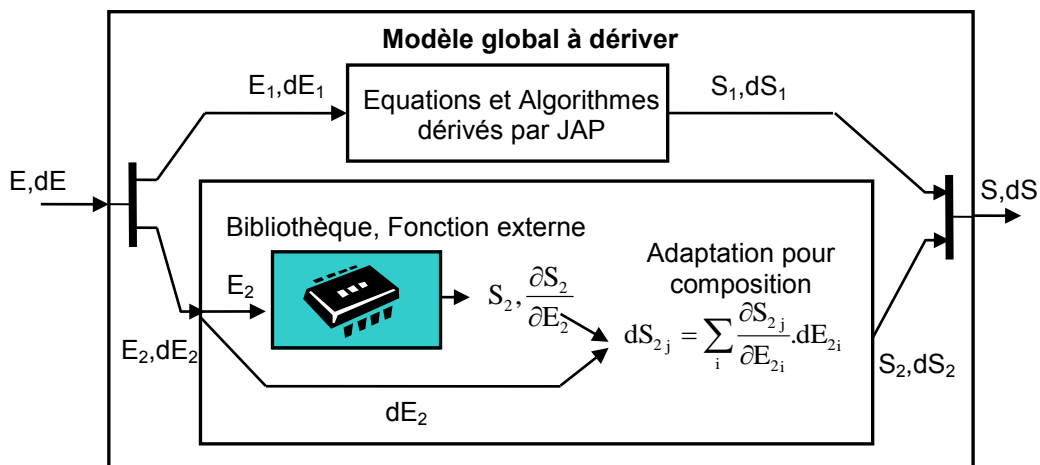


Figure III-13. Composition pour des sous fonctions externes

III.3.2. Architecture de JAP

Avec le principe présenté ci-dessus de simulation de surcharge d'opérateurs, nous avons choisi d'implémenter la première version de JAP en mode direct pour sa facilité de mise en œuvre.

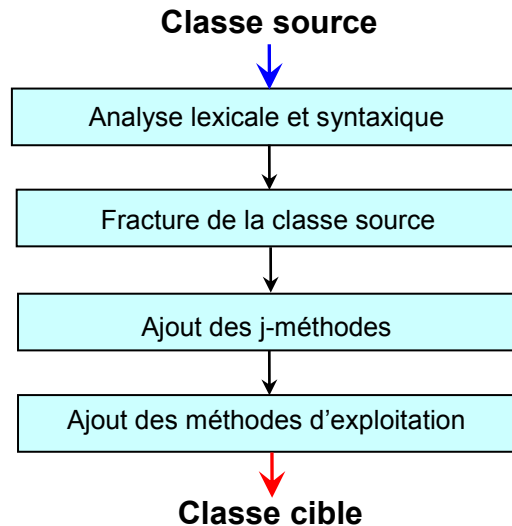


Figure III-14. Architecture de JAP

JAP est un compilateur informatique qui traite un fichier de texte Java et génère automatiquement le code de dérivation. La Figure III-14 montre l'architecture de JAP qui se décompose en quatre étapes. Par rapport au principe général de technique de génération de code présenté dans la section III.2.2.2. , illustré dans la Figure III-4, JAP ne génère pas de code intermédiaire, il génère directement le code dérivatif à partir du code source.

a) Analyse lexicale et syntaxique

L'entrée de JAP est une classe source à dériver (un fichier de texte Java). Cette classe peut contenir plusieurs méthodes. Les bibliothèques de calcul (.jar) sont mises dans le répertoire */lib* au même niveau que la classe à dériver comme le montre la Figure III-15.

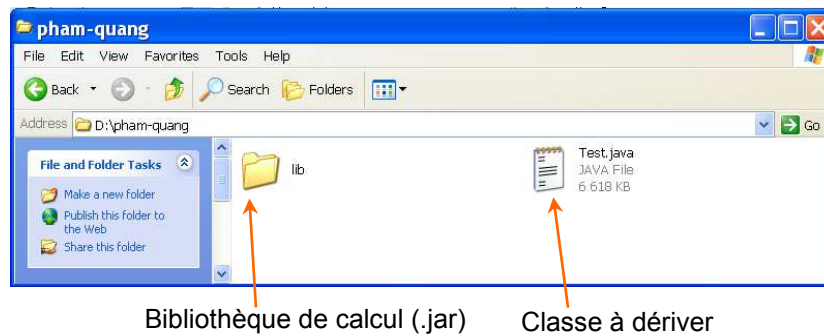


Figure III-15. Bibliothèques de calcul (.jar) de la classe à dériver

Dans cette première étape, JAP utilise une bibliothèque de JavaCC (Java Compiler Compiler, <http://javacc.java.net/>) pour la vérification lexicale et syntaxique de la classe source suivant la norme de Java VM version 1.1. S'il y a une erreur lexicale ou syntaxique (s'il manque le « ; » par exemple), JAP fait remonter l'erreur et la ligne correspondante à l'utilisateur.

b) Fracture de la classe source

```

1  package test;                                     //déclaration de package
2  import extf.Extfunc;                             //bibliothèques d'importation
3  /** @author pham-quang
4   * date 21 nov. 2010
5   */
6  public class Test {                               //déclaration de la classe
7      static final double Vb=400;                 //déclaration de variables globales
8
9      public double func1(double a1,double[]av){//signature de la méthode
10         double res;                             //déclaration de variables locales
11         int i;
12
13         //Cœur de calcul de la méthode
14         res = func2(av[0])+ Extfunc.extfunc(a1,av);
15         ...
16         return res*2;
17     }
18
19     public double func2(double a1) {              //signature de la méthode
20         double res;                             //déclaration de variables locales
21
22         //coeur de calcul de la méthode
23         res = Math.sin(a1/Vb);
24         ...
25         return res;
26     }
27 }

```

Figure III-16. Fracture de la classe source

JAP utilise aussi un parseur généré par JavaCC, pour ignorer les commentaires, et pour fractionner la classe source en des parties différentes comme expliqué par les commentaires dans la Figure III-16 :

- ✓ Déclaration de package
- ✓ Bibliothèques d'importation
- ✓ Déclaration de classe
- ✓ Déclaration de variables globales
- ✓ Méthodes de calcul

Chaque méthode de calcul (voir les commentaires dans la Figure III-16), est aussi fractionnée en 3 parties, par convention (voir Jap Exception 06 de l'annexe A) les variables locales doivent être déclarées juste après la signature de la méthode :

- ✓ Signature de la méthode
- ✓ Déclaration de variables locales
- ✓ Coeur de calcul

c) Ajout des j-méthodes

JAP ajoute pour chaque méthode de calcul, la *j-méthode* associée comme illustré précédemment dans la section III.3.1.2.

- ✓ Signature de la méthode : le type *double* est remplacé systématiquement par *Jdouble1*, le nom de la méthode est préfixé par « *j* ».
- ✓ Déclaration de variables locales : le type *double* est systématiquement remplacé par *Jdouble1*.
- ✓ Cœur de calcul : toutes les opérations et fonctions intrinsèques sont transformées en des méthodes statiques de la classe *Jdouble1*.

d) Ajout des méthodes d'exploitation

Pour chaque méthode « *public* » de calcul, JAP ajoute trois méthodes d'exploitation comme montrée dans la section III.3.1.3.

III.3.3. Limitations de JAP

Le langage Java est un langage de programmation orienté objet. Compte tenu de la complexité de ce langage, JAP ne peut pas dériver tous les codes Java. Il faut préciser les possibilités et les limites de ce dérivateur :

✓ L'entrée de JAP est une classe (un fichier texte) qui contient plusieurs méthodes de calcul en Java. JAP ne peut pas traiter un fichier qui contient plusieurs classes, de l'héritage, etc.

✓ JAP peut traiter des bibliothèques de calcul (*.jar*). Une bibliothèque de calcul doit fournir le calcul du Jacobien des fonctions qu'elle contient.

✓ JAP dérive une classe de calcul écrit en langage Java, mais cette classe doit respecter les conventions imposées par JAP. La liste des exceptions de JAP liées à ces conventions est détaillée dans l'annexe A. Si une des conventions n'est pas respectée, un message d'erreur apparaît lors du processus de génération de JAP comme le montre la Figure III-17.

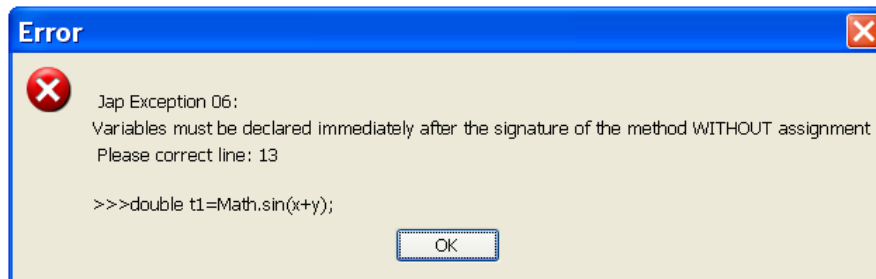


Figure III-17. Un message d'erreur de JAP

III.4. Performances de JAP

Pour évaluer les performances de JAP, nous avons comparé JAP avec un outil existant reconnu de DA - ADOL-C [WAL] sur l'algorithme de calcul de déformation de la poutre qui a été détaillé au chapitre II. L'outil ADOL-C est choisi parce qu'il est une des références dans le monde de la DA, qu'il a été déjà exploité à plusieurs reprises dans notre équipe au G2Elab [FIS-04] [ENC-09-2], et qu'il est implémenté dans notre framework CADES.

Le Tableau III-1 caractérise le contenu de l'algorithme de calcul de déformation mécanique de la poutre. On voit aisément que cet algorithme est assez complexe, car il contient la plupart des structures de contrôle proposées par les langages de programmation. On peut noter que c'est un modèle vectoriel, les arguments des méthodes sont de type *double*, *double[]*, *double[][]*, *int*.

Nombre de lignes de code	≈930
Nombre de méthodes	4
Nombre d'instructions <i>if-then-else</i>	57
Nombre d'instructions <i>for</i>	56
Nombre d'instructions <i>while</i>	4

Tableau III-1. Contenu de l'algorithme de calcul de déformation de la poutre

L'outil de DA ADOL-C est utilisé pour évaluer le gradient de cet algorithme en C++ et JAP pour le même algorithme en Java. Comme le mode inverse n'est pas implémenté dans JAP, on ne compare que le mode direct.

Nous trouvons que les gradients calculés par les deux outils sont identiques. Nous pouvons donc en conclure que notre implémentation est fonctionnelle. Ensuite, nous avons souhaité mesurer les performances en temps et en mémoire. Malheureusement nous ne sommes pas parvenus à quantifier clairement l'occupation mémoire propre à la partie de dérivation de code.

La géométrie de la poutre est montrée sur la Figure III-18. Pour faire varier la quantité de calcul, on fait varier le nombre de forces (NF) appliquées sur la poutre. Pour chaque calcul, *nGrad* est le nombre de variable d'entrée à dériver du modèle.

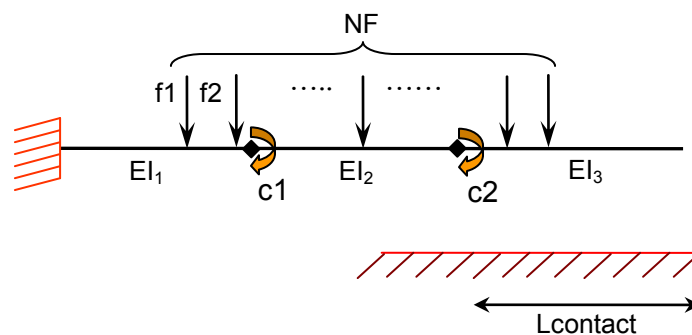


Figure III-18. Géométrie de la poutre dont on calcule la déformation

Le Tableau III-2 montre la comparaison des performances en temps CPU entre ADOL-C et JAP en mode direct. On peut observer que la fonction originale en C++ est plus rapide que la version Java. Ce qui est généralement le cas entre un langage compilé directement en code machine et un langage interprété comme Java. C'est la raison pour laquelle nous ne comparons pas la valeur absolue, mais plutôt le facteur calculé par Eq. III-9. C'est le facteur entre le temps d'évaluation des dérivées par rapport à un paramètre, et le temps d'évaluation de la fonction originale.

$$\text{Facteur} = \frac{1}{n\text{Grad}} \cdot \frac{\text{temps_d'évaluation_du_Jacobien}}{\text{temps_d'évaluation_de_la_fonction_originale}} \quad \text{Eq. III-9}$$

On peut observer dans le Tableau III-2 que le facteur d'ADOL-C et celui de JAP sont du même ordre de grandeur. Ce qui est très rassurant quant à notre implémentation vis-à-vis d'un outil comme ADOL-C sur lequel se penchent de nombreux informaticiens et numériciens.

	NF=50 nGrad=113	NF=75 nGrad=163	NF=100 nGrad=213
C++ (fonction originale) [s]	0,140	0,188	0,234
Java (fonction originale) [s]	0,250	0,343	0,438
ADOL-C (jacobian) [s]	49,03	93,86	147,8
JAP (jacobian) [s]	106,6	251,2	371,3
Facteur ADOL-C	3,10	3,06	2,97
Facteur JAP	3,77	4,49	3,98

Tableau III-2. Comparaison des performances en temps CPU entre ADOL-C et JAP en mode direct

De plus, JAP permet d'évaluer les dérivées de manière sélective pour économiser des calculs inutiles. Le Tableau III-3 montre l'avantage en temps CPU de l'évaluation des dérivées sélectives, dérivées par rapport à quelques paramètres d'entrée (nGrad=1, 4, 10) par rapport à l'évaluation de tout le Jacobien.

	NF=50	NF=75	NF=100
nGrad=1	3,89	5,77	7,58
nGrad=4	6,12	9,19	11,73
nGrad=10	9,75	14,42	19,28
Tout le Jacobien	106,6	251,2	371,3

Tableau III-3. Performances en temps CPU en [s] de JAP entre l'évaluation de tout le Jacobien et l'évaluation de dérivées sélectives

III.5. Usages de JAP

III.5.1. Mise en oeuvre de JAP dans CADES

III.5.1.1. Description d'un modèle de dimensionnement dans CADES

Pour bien comprendre la mise en œuvre de JAP dans CADES, ce paragraphe présente brièvement le langage *sml*¹ pour décrire un modèle mathématique de dimensionnement dans ce framework. Le langage *sml* est basé sur la description des modèles analytiques explicites. Il est aussi capable de décrire des algorithmes numériques pour l'implémentation des aspects mathématiques plus complexes.

a) La modélisation analytique

La modélisation analytique se réalise dans le langage *sml* en utilisant des équations analytiques explicites avec la syntaxe illustrée dans la Figure III-19.

```
sml :  
nom_variable=expression_mathematique ;
```

Figure III-19. La syntaxe d'une équation du langage *sml*

Le membre du gauche est une variable qui sera affectée du résultat de l'évaluation de l'expression mathématique située à droite du signe égal.

La syntaxe du langage *sml* permet aussi de définir et d'utiliser des fonctions. Ces fonctions sont appelées *internes* puisqu'elles sont définies et exploitées à l'intérieur du modèle uniquement

```
sml :  
nom_fonction(argument1,...)=expression_mathematique ;
```

Figure III-20. La syntaxe d'une fonction interne du langage *sml*

Avant nos travaux sur JAP, deux outils de génération pour ce modèle analytique étaient disponibles :

- ✓ *JavaGenerator* : génère du code cible en Java. Le Jacobien du modèle est évalué symboliquement. La limite de ce compilateur est qu'il n'a été implémenté que pour des fonctions et équations scalaires.

¹ *sml* : System Modelling Language

✓ *AdolCGenerator* : utilise le dérivateur automatique ADOL-C [WAL] en C/C++ pour dériver des modèles pouvant exploiter des fonctions et équations vectorielles. Comme les générateurs de CADES produisant des modèles exécutables en Java, une interface (JNI¹) entre Java et le langage natif créé par le code C/C++ a dû être mise en œuvre. Cette connexion constitue la principale limite de ce générateur.

Avec la mise en œuvre de JAP dans CADES, il est maintenant possible de générer un code 100% Java à partir d'équations et fonctions scalaires et vectorielles.

b) La modélisation semi-analytique en utilisant des fonctions externes

Dans la modélisation, certains modèles ne peuvent pas être décrits seulement sous la forme analytique par des équations, mais plutôt sous la forme semi-analytique par l'utilisation d'algorithmes numériques. Ces algorithmes peuvent implémenter des méthodes numériques génériques (comme par exemple l'intégration des fonctions, des algorithmes de résolution des systèmes d'équations implicites non linéaires, etc.) ou des méthodes numériques dédiées (comme par exemple l'algorithme de calcul de déformation de la poutre pour le nano commutateur magnétique, décrite dans le chapitre II). Le langage *sml* donne au concepteur la possibilité d'intégrer aux équations et fonctions définissant le modèle analytique, des fonctions de bibliothèques externes décrites en Java, si toutefois elles fournissent également leur Jacobien.

Pour illustrer ce besoin, reprenons l'exemple de la thèse de P. Enciu [ENC-09-2] donné sur la Figure III-21 qui calcule la résistance d'un fil électrique et la chute de tension à ses bornes. La résistance n'est pas définie par une fonction analytique, mais à partir d'une base de données obtenue par mesures, interpolées par un code numérique. Ici la fonction **Interpolate** s'appelle « fonction externe », car elle n'est pas décrite sous forme analytique, mais appartient à une bibliothèque Java.

¹ JNI : Java Native Interface

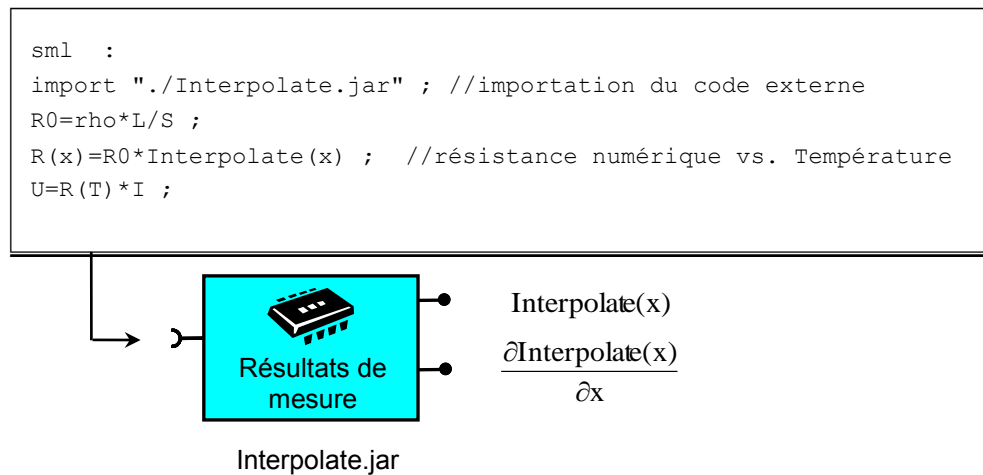


Figure III-21. Modélisation semi-analytique dans le langage sml en utilisant des fonctions externes

Le calcul des dérivées de la fonction externe doit être associé au code de calcul dans la bibliothèque. De tels Jacobiens peuvent être obtenus par dérivation symbolique en exploitant les propriétés mathématiques des calculs implémentés dans le code. C'est par exemple le cas pour la dérivation d'un programme de calcul d'une intégrale numérique, ou celui de résolution de fonctions implicites (théorème des fonctions implicites) etc.

En général, on peut calculer des dérivées par la méthode de différences finies, mais avec une précision limitée, la technique de DA est alors une solution exploitable. Pendant la thèse de P. Enciu [ENC-09-2], un générateur du Jacobien de telles bibliothèques de fonction a été réalisé pour CADES en utilisant ADOL-C. Ce générateur, nommé *ADOLcExternalFunctionGenerator*, propose au concepteur d'écrire son algorithme en langage ADOL-C (du C légèrement modifié, voir [ENC-09-2]), et crée automatiquement la bibliothèque contenant le code de calcul en C et son Jacobien associé. Deux limites apparaissent dans cette approche, la première liée à la nécessité de connaître la syntaxe d'un programme ADOL-C, la seconde au fait d'ajouter une couche d'interface Java/C entre chaque appel à cette fonction externe. Ce dernier point est très pénalisant vis-à-vis des performances (temps de calcul et mémoire).

Avec la mise en œuvre de JAP, la description du code de calcul est faite en pur Java (sans annotation ou utilisation de type particulier), et l'interfaçage avec le modèle analytique est rapide, car dans un langage unique.

III.5.1.2. JAP_ExternalFunctionGenerator : Générateur de fonctions externes algorithmiques

JAP_ExternalFunctionGenerator est un générateur de CADES qui utilise directement JAP pour dériver des algorithmes numériques écrits en Java. La Figure III-22 montre l'architecture de ce générateur.

Le processus de génération se décompose de trois étapes :

- ✓ Génération de code de dérivation : fait appel à JAP pour générer le code de dérivation à partir d'un fichier texte (.java).
- ✓ Compilation : compile du code généré par le compilateur Java
- ✓ Packaging : encapsule toutes les ressources compilées dans une archive Java (.jar) pour constituer la librairie de fonctions qui seront appelées dans le modèle analytique.

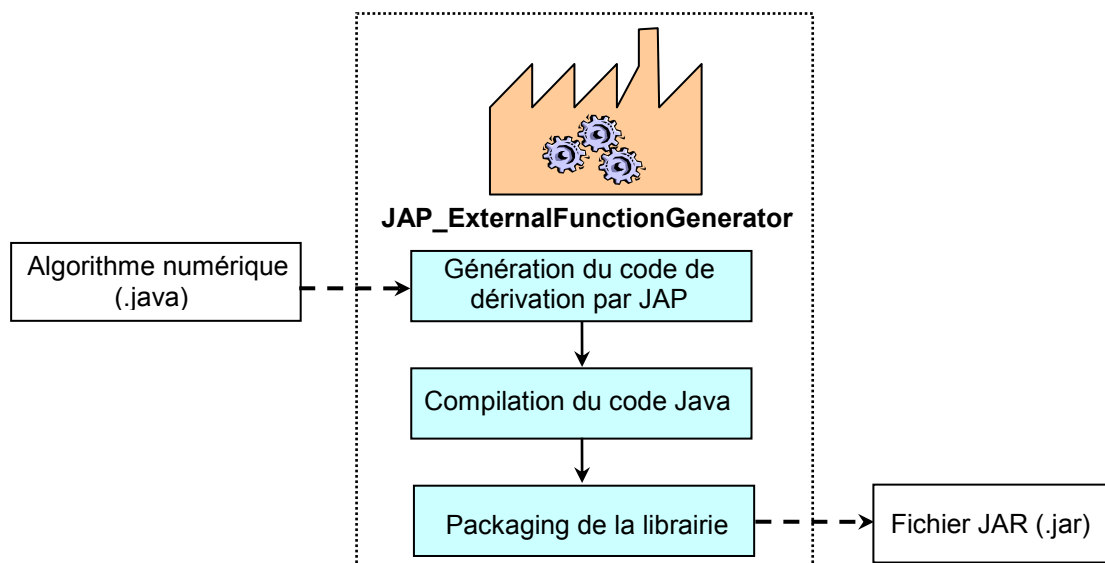


Figure III-22. Architecture de JAP_ExternalFunctionGenerator

La Figure III-23 montre l'IHM et de ce générateur de CADES pour des fonctions externes.

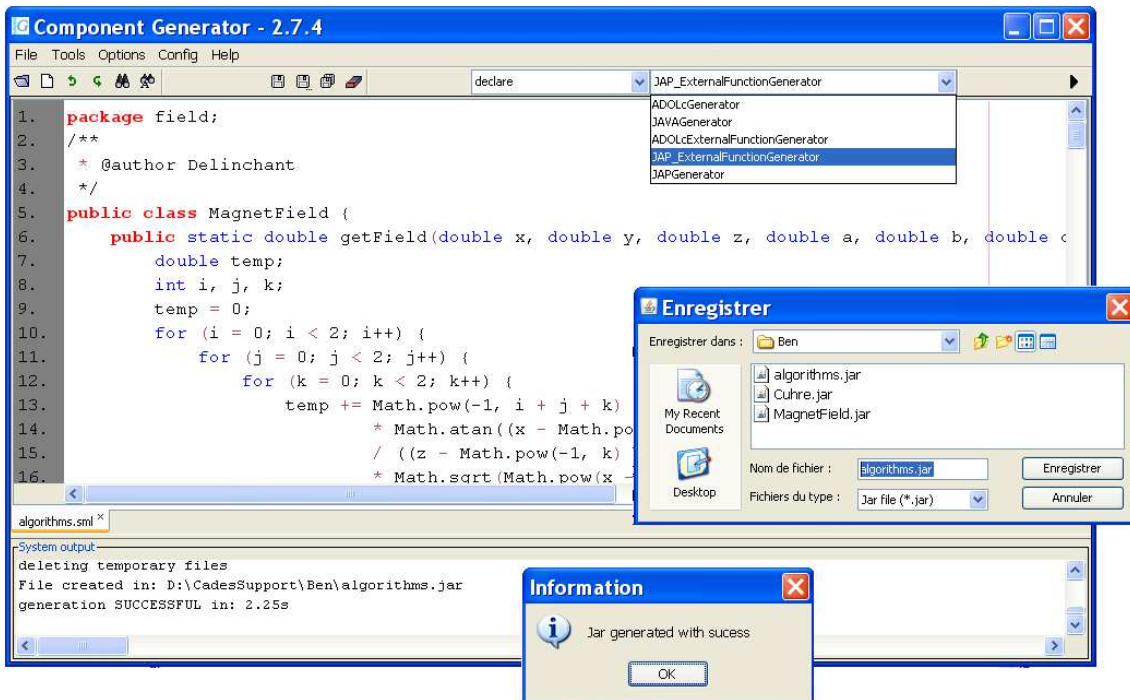


Figure III-23. IHM de JAP_ExternalFunctionGenerator

III.5.1.3. JAPGenerator : Générateur de modèle de dimensionnement sml

La Figure III-24 montre l'architecture de *JAPGenerator*, une utilisation de JAP pour dériver des modèles de dimensionnement *sml* dans CADES. Le *JAPGenerator* agit avec la même philosophie qu'un compilateur de langage informatique. Ainsi, le modèle de dimensionnement écrit dans le langage source *sml* est transformé dans le langage cible du composant de calcul suivant la norme ICAR de CADES (un fichier *.icar* est semblable à une archive Java).

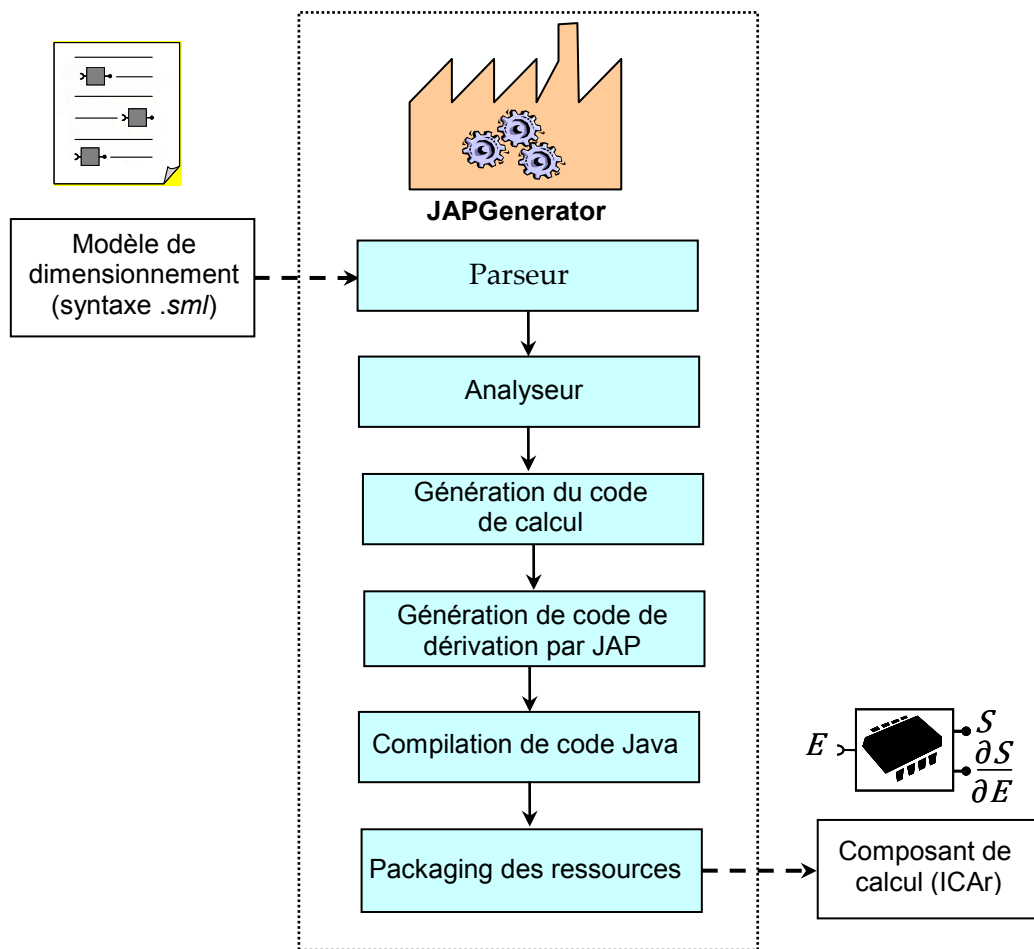


Figure III-24. Architecture de JAPGenerator

La signification des étapes de génération est la suivante :

- ✓ Parseur : effectue l'analyse lexicale et syntaxique du modèle mathématique de dimensionnement en s'appuyant sur un jeu de règles grammaticales du langage *sml*. Ce parseur est factorisé pour les générateurs ADOL-C et JAP
- ✓ Analyseur : effectue l'analyse sémantique en s'appuyant sur les informations données par le parseur. Les variables d'entrées/sorties, les équations ordonnancées, la cohérence du modèle contrôlée (appels de fonctions existantes, pas de bloc implicites, etc.) sont identifiées.
- ✓ Génération du code de calcul : les équations ordonnancées sont générées en Java.
- ✓ Génération du code de dérivation : utilise JAP pour dériver le code de calcul du modèle.

✓ Compilation : compile le code source (calcul du modèle et des dérivées) avec un compilateur Java.

✓ Packaging : encapsule toutes les ressources compilées dans une archive Java pour faire un composant de calcul suivant la norme ICAr de CADES.

III.5.2. Exemples d'utilisation de JAP dans CADES

III.5.2.1. Validation et performances de JAP dans CADES

Afin de tester la robustesse de JAP, mais également ses performances dans différentes conditions, nous avons repris les cas tests de CADES, pour lesquels nous avons des solutions de référence d'optimisation obtenues par le générateur ADOL-C. Les spécificités de ces modèles sont montrées dans le Tableau III-4.

Applications	Nombre d'équation	Fonctions externes
Machine à concentration de flux [SEN-11]	~400	Solveur d'implicites
Actionneur linéaire [CHI-01]	~300	Solveur d'implicites
Alternateur à griffe [ALB-04]	~600	non
Application du projet MEGEVE [NGU]	~370	Algorithmes
Nano commutateur magnétique (chapitre II)	~70	Force, couple magnétique Module mécanique Solveur d'implicites

Tableau III-4. Descriptions des modèles des applications de test

Les tableaux suivants montrent la comparaison des résultats obtenus par les outils *JAPGenerator* et *AdolCGenerator* de CADES en différents termes :

✓ Taille du code généré : c'est la taille du composant de calcul (ICAr).

✓ Temps de génération : temps nécessaire pour générer le composant de calcul (ICAr) à partir du modèle *sml*.

✓ Temps d'optimisation : l'algorithme utilisé est SQP.

De plus, JAP autorisant le choix des grandeurs à dériver, ce qui est particulièrement intéressant pour les modèles exploitant des codes numériques. Les temps d'exploitation de cette fonctionnalité sont également comparés.

Les cinq tests ont été réalisés et les résultats se trouvent en annexe D. L'analyse de ces résultats nous conduit aux commentaires suivants :

Commentaires :

✓ *JAPGenerator* et *AdolCGenerator* convergent vers les mêmes résultats d'optimisation avec le même nombre d'itérations. Ce qui valide la robustesse et la justesse du calcul du Jacobien de JAP dans CADES.

✓ *JAPGenerator* montre l'avantage par rapport à *AdolCGenerator* en terme de taille de composant de calcul (ICAr) et du temps de génération. Ceci s'explique par la complexité introduite par le langage C et son interfaçage avec Java dans *AdolCGenerator*.

✓ En terme de temps d'optimisation : sur les petits modèles, le temps d'optimisation est de quelques secondes, on ne voit donc pas la différence entre les deux générateurs. Mais sur le modèle complexe de nano commutateur magnétique avec 3 fonctions externes, le *JAPGenerator* montre son intérêt (5 fois plus rapide) car il génère le modèle purement en Java, il ne perd pas de temps d'interfaçage entre les différents langages.

✓ Concernant l'option de sélectivité de JAP : sur les petits modèles, cette option n'est pas utile. Par contre, elle se révèle très intéressante, voire indispensable pour des modèles de calcul plus lourds. Un facteur de gain de 3 est ainsi obtenu pour notre application du nano commutateur magnétique. La sélectivité est d'ailleurs une option qui a été introduite dans le générateur *JavaGenerator* (dérivation symbolique) pour les modèles produits par MacMMems [RAK-07]. En effet, ceux-ci utilisent des intégrations numériques et certaines optimisations complexes peuvent atteindre l'heure.

III.5.2.2. Modélisation et optimisation du système de suspension magnétique

a) Description de l'application

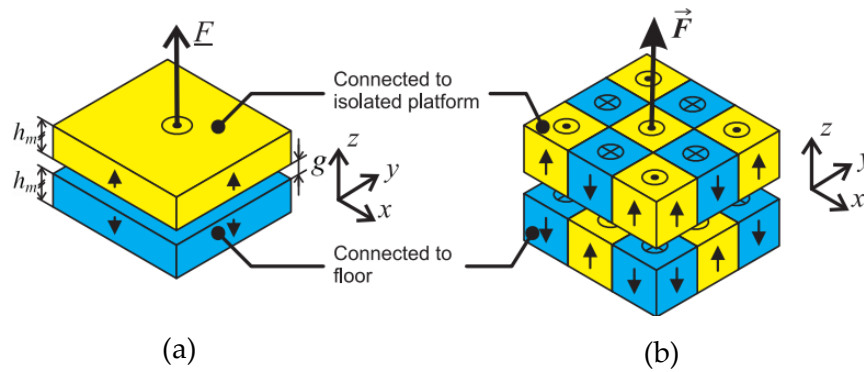


Figure III-25. Représentation schématique du système de suspension magnétique (a) à deux aimants cubiques, (b) à des matrices d'aimants (extrait de [JAN-11])

Le système de suspension magnétique qui se compose de deux aimants cubiques ou des matrices d'aimants comme illustré par la Figure III-25. Ce système joue le rôle d'un actionneur pour stabiliser des systèmes intrinsèquement instables. Pour les détails de cette application le lecteur peut consulter [JAN-09], [JAN-11].

b) Modélisation dans MacMMems/CADES sans JAP

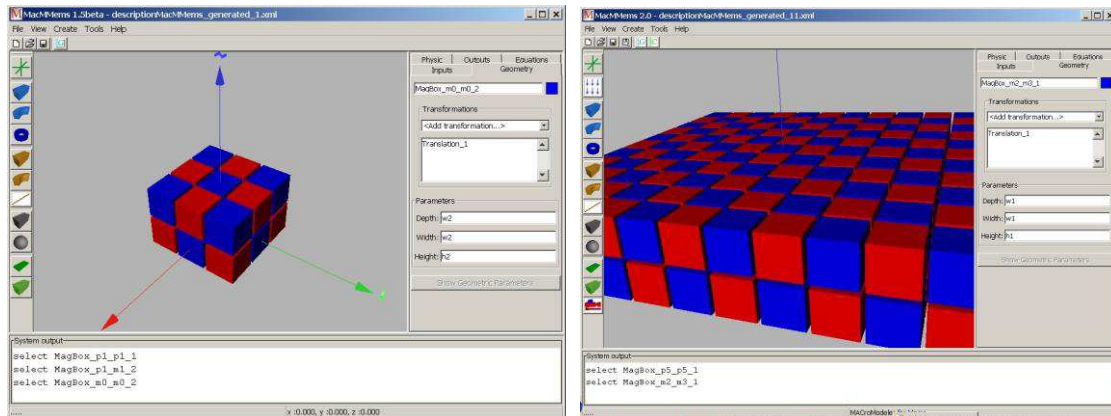


Figure III-26. Modélisation du système de suspension magnétique dans MacMMems

La modélisation des matrices d'aimant du système de suspension magnétique dans MacMMems est montrée sur la Figure III-26. Avant la disponibilité de JAP, la modélisation de chaque cube était indépendante : chaque aimant produit un champ qui est décrit par une fonction, et chaque force est obtenue par l'intégration des champs produits par tous les aimants.

La difficulté n'est pas la description, car celle-ci peut être automatisée par un script comme celui produisant la structure 11x11 ci-dessus. La difficulté vient dans le modèle généré pour cette structure :

- ✓ Fichier d'équations : 14 Mo avec 200000 lignes (équations générées en 12 secondes).
- ✓ Analyse des équations pour la génération du code de calcul : impossible! car la taille limite est dépassée (limite de mémoire de Java).

c) Modélisation avec algorithme dans MacMMems/CADES grâce au JAP

Grâce à JAP, le calcul du champ produit par chaque aimant est décrit par un algorithme plutôt que par de nombreux appels de fonction. Le champ total est calculé par une sommation de tous les champs par l'écriture d'une double boucle, plutôt que par la génération du code des N×N appels aux fonctions.

L'avantage de la modélisation par algorithme est l'optimisation du code généré par rapport à la modélisation purement par équations. Le Tableau III-5 montre cet avantage en terme de taille de fichier modèle (*sml*), de temps de génération et de taille du composant de calcul (ICAr).

	Modèle 3×3		Modèle 5×5	
	équations	équations+ algorithme	équations	équations+ algorithme
Taille de fichier modèle (<i>sml</i>)	100 ko	1ko + 5ko	500 ko	5ko + 5ko
Temps de génération du composant de calcul	15 s	6 s	4 min	8 s
Taille du composant de calcul (ICAr)	150 ko	100ko + 26ko	300 ko	110ko + 26ko

Tableau III-5. Avantage de la modélisation avec algorithme sur le modèle de système de suspension magnétique

Commentaires :

- ✓ En terme de taille de fichier modèle (*sml*), la modélisation avec algorithme montre un avantage avec une partie fixe (5ko) d'algorithme et une partie d'équations beaucoup plus petite par rapport à celle initiale sans JAP.

✓ Comme la taille du fichier modèle est petite, on gagne énormément en terme de temps de génération. C'est la raison pour laquelle pour de gros modèles (11×11), la modélisation avec algorithme est indispensable.

d) Résultats d'optimisation

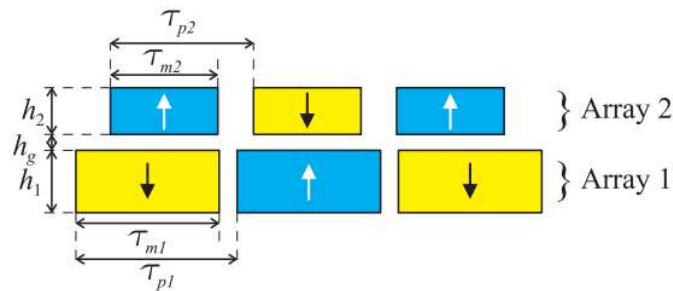


Figure III-27. Dimensions utilisées dans l'optimisation pour une topologie où le nombre d'aimants sur les côtés est égal à 3 (n=3)

Les objectifs de l'optimisation sont de maximiser la force et de minimiser la fréquence de résonance pour chaque topologies (3, 5, 7...). La Figure III-27 montre les paramètres à dimensionner pour une topologie où le nombre d'aimants sur les côtés, n est égal à 3. Pour plus de détails sur ce cahier des charges, le lecteur peut consulter [JAN-11].

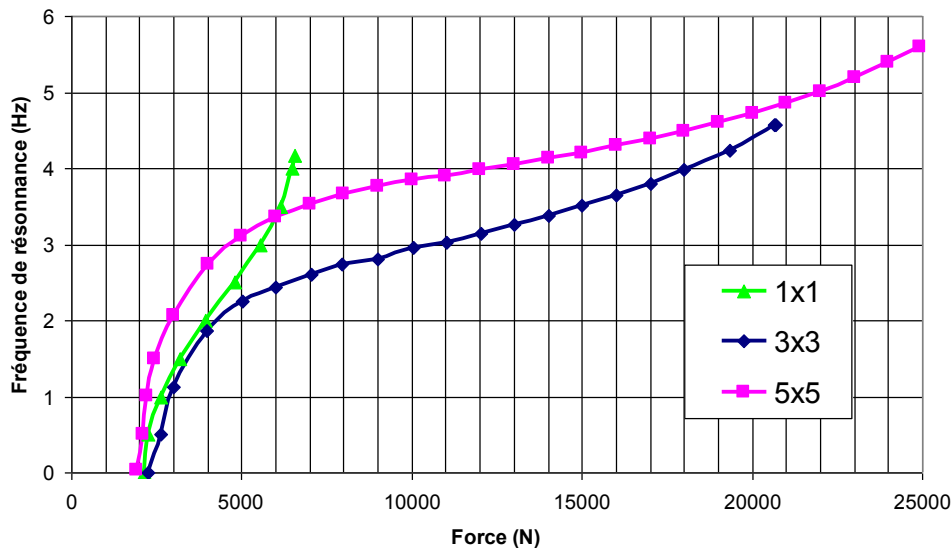


Figure III-28. Front de Pareto entre la force et la fréquence de résonance avec 3 structures (1×1, 3×3, 5×5)

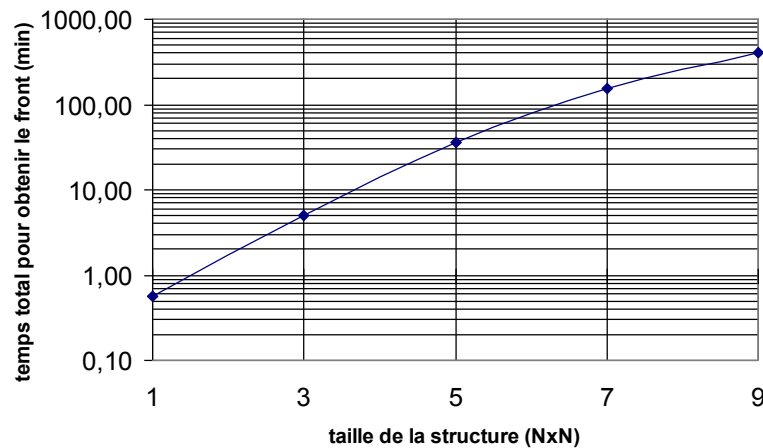


Figure III-29. Temps pour obtenir le Front de Pareto en fonction de la structure

Ces résultats montrent tout l'intérêt d'utiliser des outils dédiés comme MacMMems associé à JAP, pour concevoir des systèmes magnétiques tels que ces suspensions passives. Un tel système montre la limite entre des descriptions purement analytiques, et l'utilisation de méthodes complètement numériques (éléments finis par exemple). Ici, MacMMems associé à JAP permet de décrire un modèle semi analytique parfaitement adapté au système modélisé.

III.5.2.3. Utilité d'un code 100% Java

Nous venons de voir que JAP nous ouvrait la voie vers des modèles complexes (semi-analytiques) 100% Java. Cette fonctionnalité a été exploitée pour permettre de distribuer le composant de calcul (ICAr), généré à partir de PC sous windows, sur un serveur web externalisé fonctionnant sous Linux. Ceci est d'abord possible grâce au fait qu'un ICAr est un composant logiciel, avec toutes les propriétés informatiques que cela implique (autonomie, auto descriptif, ...), ensuite, grâce au langage Java dont nous avons mis en œuvre la dérivation automatique avec JAP. Dans cette expérience, liée au projet de recherche ANR SIMINTEC, nous avons réussi à optimiser un régulateur de température associé à un modèle thermique de bâtiment et à un système de chauffage, sans disposer localement du modèle. L'optimiseur allant interroger (de façon transparente) un serveur web distant, à chaque itération.

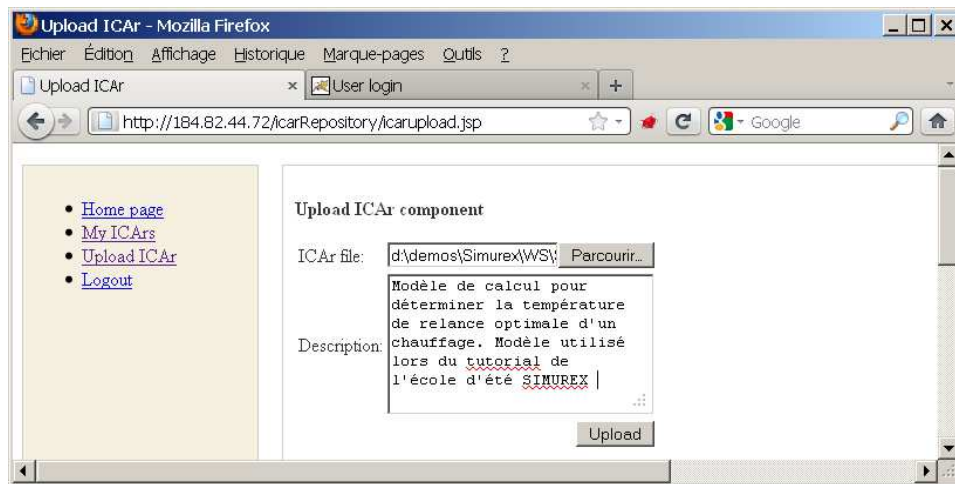


Figure III-30. Interface graphique du serveur web permettant d'uploader un composant du calcul sur le serveur de calcul

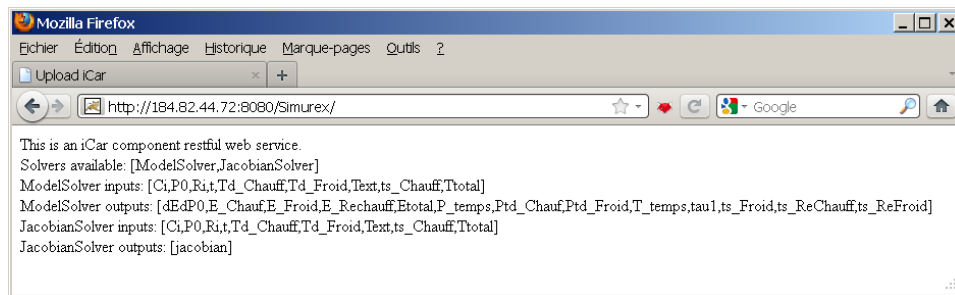


Figure III-31. Informations relatives au composant de calcul distribué sous forme de *WebService* (entrées/sorties du modèle, etc.)

L'optimisation du modèle distant a été comparée à une optimisation locale. Des problèmes de troncature de données ont été rencontrés lors de la transmission du Jacobien via le réseau (mode texte http) conduisant à une optimisation un peu moins stable (14 itérations au lieu de 10) et un temps de latence supérieur conduisant à un temps global de 3 secondes au lieu de 0.1 seconde.

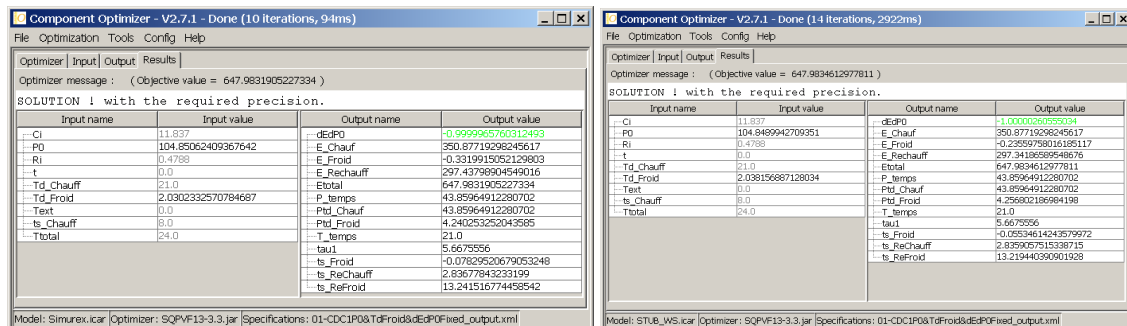


Figure III-32. Comparaison des résultats d'optimisation entre le mode local (gauche) et le mode distant via le *WebService* (droite)

III.5.2.4. Optimisation du nano commutateur magnétique

Nous présentons ici la description du modèle du nano commutateur magnétique, et la composition du Jacobien par JAP pour ce modèle. Le cahier des charges, les résultats d'optimisation de différentes stratégies seront présentées dans la section IV.2 du chapitre IV sur l'étude complète du nano commutateur magnétique.

La modélisation semi-analytique du nano commutateur magnétique a été décrite dans le chapitre II : c'est un couplage magnéto-mécanique. La description de ce couplage dans CADES comprend environ 70 équations *sml* qui font appel à des algorithmes comme des fonctions externes.

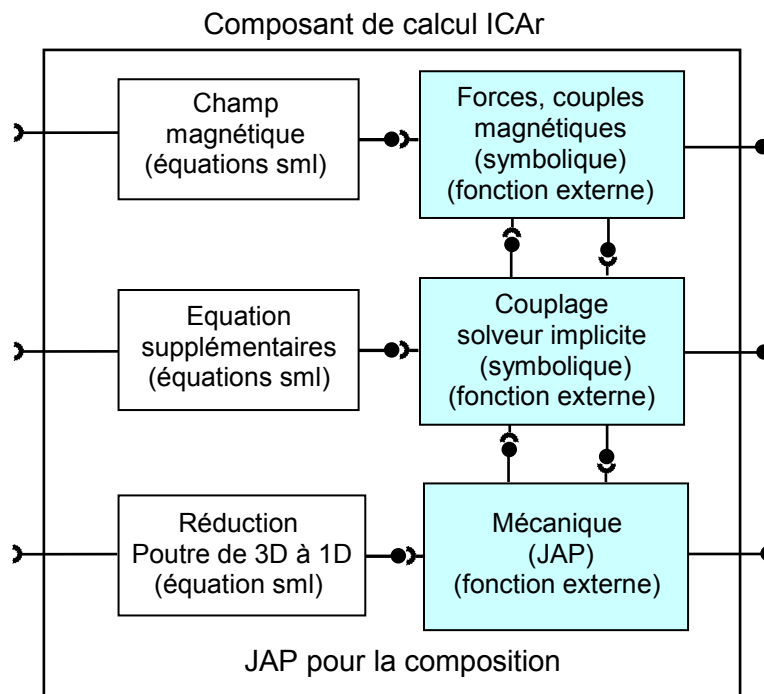


Figure III-33. La composition de Jacobien pour le nano commutateur magnétique dans CADES

La Figure III-33 montre le principe de composition du Jacobien par JAP pour ce couplage. On y retrouve 3 blocs de modèles analytiques en *sml*, ainsi que 3 blocs de fonctions externes. Ces 3 derniers correspondent aux :

- ✓ Calcul des forces et couples magnétiques générés par MacMMems (voir chapitre II) décrits pas des intégrations numériques des champs calculés analytiquement. Les fonctions externes génériques d'intégrations numériques utilisent une dérivation symbolique pouvant conduire à des calculs de sensibilité plus rapide que le calcul du modèle lui-même.

✓ Couplage du modèle magnétique et du modèle mécanique. Cet algorithme générique implémente une méthode de Gauss-Seidel pour résoudre le système implicite dont la dérivation est faite de manière symbolique en utilisant le théorème des fonctions implicites.

✓ Calcul de la déformation mécanique et de la longueur de contact. Ce modèle, spécifique à notre étude, est dérivé automatiquement par JAP.

Une fois généré, ce modèle peut être exploité dans *CadesOptimizer* et *FGOT*, afin de dimensionner le nano commutateur magnétique (chapitre IV).

III.5.3. Utilisation de JAP pour la simulation dynamique

Dans cette section, nous présentons l'utilisation directe de JAP (sans passer par *CadesGenerator*) par Franck VERDIERE (ingénieur de recherche au G2Elab) dans le cadre du projet de recherche ANR PLUMES. L'objectif est de créer l'outil *ModelicaToICAr*, un outil générateur de composants ICAr dynamiques à partir du langage standardisé Modelica [MOD]. Cet outil consiste à générer, à partir d'un modèle décrit par des DAE (équations différentielles algébriques), un code de calcul exécutable formulé en ODE (équations différentielles ordinaires). Dans ce contexte, le dérivateur automatique JAP a été utilisé pour produire le Jacobien des DAE, comme nous allons l'expliquer ci-dessous.

a) *Modèle dynamique des équations différentielles ordinaires (ODE)*

Le modèle dynamique ODE d'ordre N (en se ramenant à un système de N équations du premier ordre) est formulé explicitement par une équation vectorielle algébrique linéaires ou non linéaires définissant les sorties du modèle en fonction du vecteur d'état :

$$\begin{cases} \dot{X}(t) = f(X(t), U(t), p, t) \\ Y(t) = g(X(t), U(t), p, t) \end{cases} \quad \text{Eq. III-10}$$

où X : vecteur d'états, U : vecteur des sources, p : les paramètres, t : la variable temps, Y : vecteur des sorties.

En particulier, les composants ICAr dynamiques sont à même de définir des modèles linéaires tels que :

$$\begin{cases} \dot{X}(t) = A \cdot X(t) + B \cdot U(t) \\ Y(t) = C \cdot X(t) + D \cdot U(t) \end{cases} \quad \text{Eq. III-11}$$

où A, B, C, D sont les matrices définissant le système dynamique linéaire.

Cette définition du modèle dynamique ODE est classique et normalisée. Elle est adoptée dans les environnements de simulation tels que Simulink ou par les environnements traitant du langage Modelica (Dymola, AMESim, OpenModelica, etc.).

b) Modèle dynamiques des équations différentielles algébriques (DAE)

Les modèles dynamiques DAE ou bien « ODE implicite avec contraintes » sont formulées sous la forme :

$$\begin{cases} f(\dot{X}(t), X(t), U(t), p, t) = 0 \\ g(Y(t), X(t), U(t), p, t) = 0 \end{cases} \quad \text{Eq. III-12}$$

Ces modèles sont utilisés principalement par les simulateurs de circuits ou par le langage de description de modèles en VHDL-AMS dans les simulateurs tels que Simplorer, PSpice, Portunus, SMASH, etc.

Les fonctions implicites inhérentes à une modélisation DAE sont généralement résolues par des solveurs itératifs tels que Newton-Raphson exploitant le Jacobien du modèle.

c) Projection DAE vers ODE

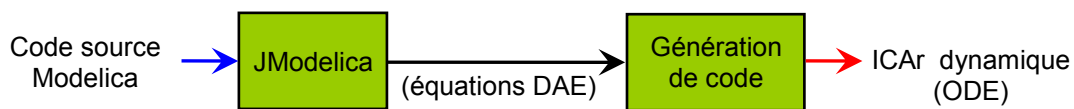


Figure III-34. Principe de fonctionnement de ModelicaToICAr

La Figure III-34 montre le principe de fonctionnement de l’outil *ModelicaToICAr*, qui comprend deux étapes :

- ✓ Transformation du modèle Modelica en équations DAE par JModelica [MOD].
- ✓ Génération de code ODE :
 - Projection des équations DAE en code Java
 - Dérivation du code Java par JAP
 - Explicitation des équations différentielle par Newton-Raphson

d) Exemple

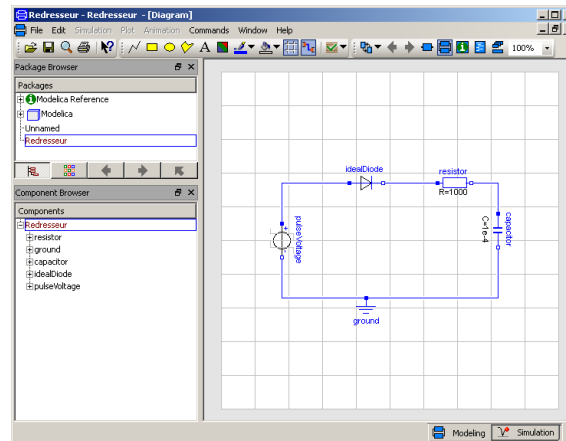


Figure III-35. Exemple de test : modèle d'un redresseur dans Dymola

Considérons l'exemple d'un circuit décrit dans Dymola [DYM] (langage Modelica) comme illustré la Figure III-35. La Figure III-36 montre la tension aux bornes du condensateur simulé respectivement dans Dymola et dans *CadesCalculator*.

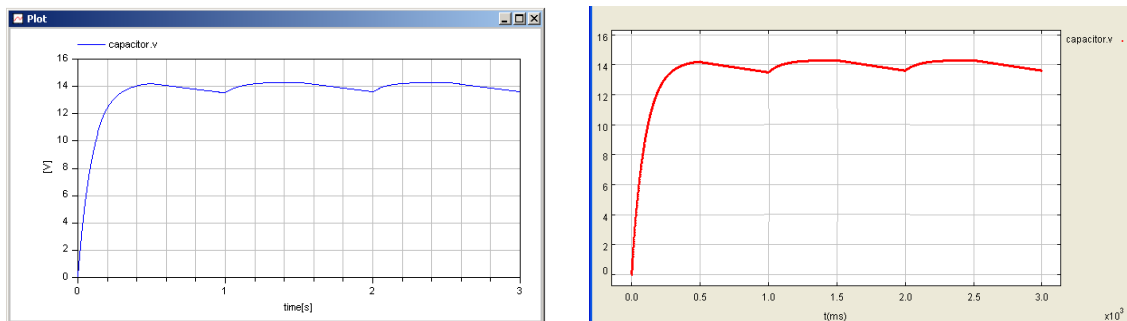


Figure III-36. Tension aux bornes de condensateur simulé dans Dymola (gauche) et *CadesCalculator* (droite)

Ainsi, JAP a été utilisé avec succès pour obtenir le Jacobien d'un modèle de calcul dynamique implicite (DAE) décrit en Java, afin de le réorienter (ODE) pour pouvoir ensuite le résoudre en exploitant des algorithmes d'intégration dynamique (Runge Kutta, etc.).

III.6. Conclusions

Ce chapitre a été consacré à une méthode de Dérivation Automatique de code en Java, et son implémentation dans l'outil JAP. Des utilisations et des applications de JAP y ont été présentées. Après avoir exprimé nos besoins spécifiques, nous avons décrit le principe et les techniques de DA et l'architecture de l'outil JAP. Nous avons aussi testé les performances et la robustesse de JAP en comparaison avec l'outil existant ADOL-C,

et les résultats obtenus ont été satisfaisants. Bien que JAP ne soit pas un outil de dérivation complet pour les programmes Java, en particulier par la non prise en compte de notions de programmation objets telles que l'héritage, il parvient déjà à répondre à un grand nombre de nos besoins. Cela a été montré en fin de chapitre par quelques usages de JAP dans les outils de dimensionnement par optimisation ainsi que dans les outils de simulation dynamique développés au G2Elab par moi-même et par d'autres utilisateurs.

Chapitre IV

Optimisation sous contraintes de fiabilité

SOMMAIRE

Chapitre IV

Optimisation sous contraintes de fiabilité	107
IV.1. Analyse de sensibilité et propagation d'incertitude.....	109
IV.1.1. Méthodes d'analyse	109
IV.1.1.1. <i>Approximation linéaire</i>	109
IV.1.1.2. <i>Méthode de Monte-Carlo</i>	112
IV.1.2. Implémentation dans CADES	112
IV.1.3. Application : analyse du nano commutateur magnétique	113
IV.2. Optimisation sous contraintes du nano commutateur magnétique	116
IV.2.1. Cahier des charges	116
IV.2.2. Stratégies d'optimisation.....	117
IV.2.2.1. <i>Optimisation directe par algorithme stochastique</i>	117
IV.2.2.2. <i>Optimisation indirecte par surface de réponse adaptative</i>	118
IV.2.2.3. <i>Optimisation par un algorithme déterministe exploitant le Jacobien</i>	119
IV.2.3. Résultats d'optimisation.....	120
IV.2.4. Sensibilité de la solution optimale	122
IV.3. Optimisation sous contraintes de fiabilité	123
IV.3.1. Contrôle de qualité : Six sigma (6σ).....	123
IV.3.2. Méthode découplée de résolution du problème d'optimisation sous contraintes de fiabilité.....	125
IV.3.3. CadesDeviationGenerator.....	127
IV.3.3.1. <i>Principe de CadesDeviationGenerator</i>	127
IV.3.3.2. <i>Composition du Jacobien dans CadesDeviationGenerator</i>	129
IV.3.4. Application pour le nano commutateur magnétique	131
IV.3.4.1. <i>Résultats d'optimisation</i>	131
IV.3.4.2. <i>Vérification des résultats par Monte-Carlo</i>	132
IV.4. Conclusions.....	134

Résumé du chapitre

Ce chapitre concerne les méthodes et les outils d'optimisation sous contraintes de fiabilité. Ces méthodes et ces outils génériques seront appliqués pour le nano commutateur magnétique. Tout d'abord, les méthodes d'analyse de sensibilité et d'analyse d'incertitude seront présentées et implémentées dans CADES. Ensuite, nous présenterons les différentes stratégies d'optimisation du nano commutateur magnétique. Nous finirons enfin par la description de l'outil CadesDeviationGenerator développé que nous utiliserons pour réaliser l'optimisation sous contraintes de fiabilité du nano commutateur magnétique.

IV.1. Analyse de sensibilité et propagation d'incertitude

IV.1.1. Méthodes d'analyse

Dans la section I.3.1 et I.3.2 du premier chapitre, les méthodes d'analyse d'incertitude et d'analyse de sensibilité ont été présentées. Dans cette section, la méthode d'approximation linéaire et la méthode de Monte-Carlo seront détaillées afin d'être implémentées dans CADES.

IV.1.1.1. Approximation linéaire

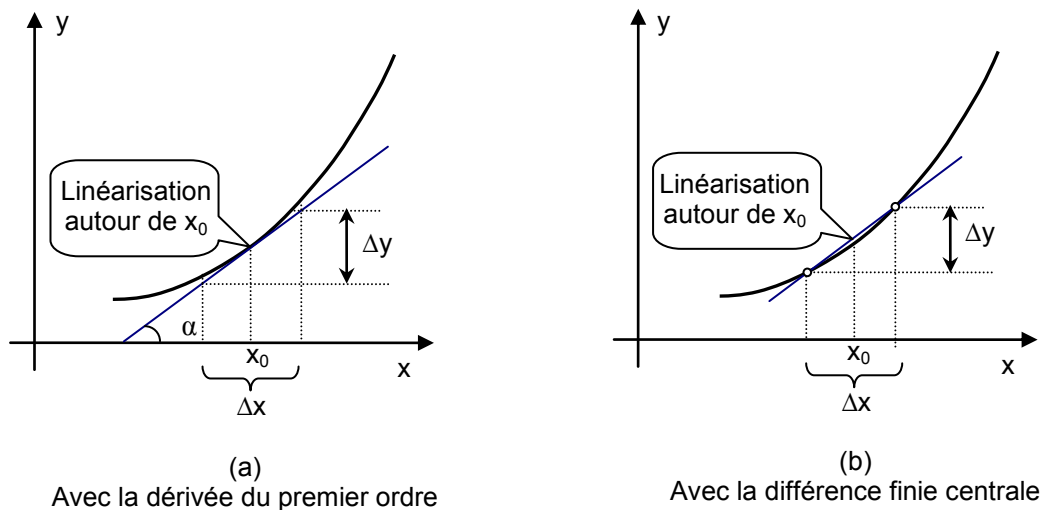


Figure IV-1. Approximation linéaire autour du point de fonctionnement

L'analyse de sensibilité par l'approximation linéaire étudie comment de petites perturbations autour du point de fonctionnement des entrées se répercutent sur la valeur de sortie. La courbe de la variation de la sortie peut-être linéarisée de deux façons comme le montre la Figure IV-1 :

- ✓ Par la dérivée partielle du premier ordre : la variation de la sortie est

calculée par :

$$\Delta y = \frac{\partial y(x_0)}{\partial x} \cdot \Delta x \quad \text{Eq. IV-1}$$

où y est la variable de sortie, x_0 représente le point de fonctionnement et Δx représente sa variation.

✓ Par différence finie : La variation de la sortie est calculée par différence finie :

- Différence finie centrale (comme illustration de la Figure IV-1-b)

$$\Delta y = y\left(x_0 + \frac{\Delta x}{2}\right) - y\left(x_0 - \frac{\Delta x}{2}\right) \quad \text{Eq. IV-2}$$

- Différence finie à droite :

$$\Delta y = y(x_0 + \Delta x) - y(x_0) \quad \text{Eq. IV-3}$$

- Différence finie à gauche :

$$\Delta y = y(x_0) - y(x_0 - \Delta x) \quad \text{Eq. IV-4}$$

La différence finie centrale donne la meilleure approximation autour du point de fonctionnement, mais en réalité il est préférable d'implémenter des différences finies à droite ou à gauche parce qu'elles demandent moins d'évaluations du modèle.

L'analyse d'incertitude s'intéresse quant à elle à la loi de probabilité de la sortie en connaissant la loi de probabilité des entrées. Dans cette thèse, on fait une hypothèse de distribution normale (loi normale Gaussienne ou loi de Laplace-Gauss) sur toutes les entrées du modèle de dimensionnement.

$$x \sim \mathcal{N}(\mu_x, \sigma_x^2) \quad \text{Eq. IV-5}$$

où μ est l'espérance (ou bien la moyenne), σ est l'écart type et donc σ^2 est la variance.

Une propriété importante et utile de la distribution normale est qu'une Gaussienne reste une Gaussienne après une transformation linéaire [SAL-00] [WIK-2]. La sortie du modèle (approximé linéairement) suit donc aussi une distribution normale :

$$y \sim \mathcal{N}(\mu_y, \sigma_y^2) \quad \text{Eq. IV-6}$$

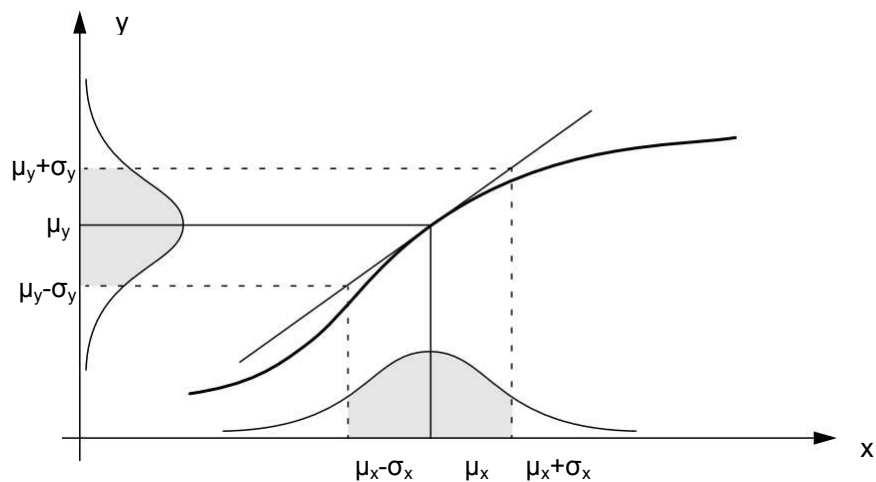


Figure IV-2. Propagation d'incertitude du modèle d'une dimension par approximation linéaire avec la dérivée de premier ordre

L'écart type de sortie est calculé comme la variation en analyse de sensibilité.

- ✓ Par l'approximation linéaire avec la dérivée partielle du premier ordre :

$$\sigma_y = \frac{\partial y(\mu_x)}{\partial x} \cdot \sigma_x \quad \text{Eq. IV-7}$$

- ✓ Par l'approximation linéaire avec différence finie (centrale par exemple) :

$$\sigma_y = y\left(\mu_x + \frac{\sigma_x}{2}\right) - y\left(\mu_x - \frac{\sigma_x}{2}\right) \quad \text{Eq. IV-8}$$

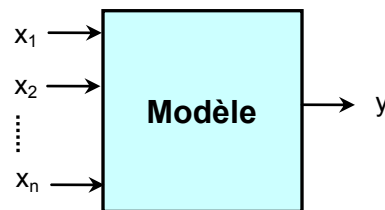


Figure IV-3. Modèle à plusieurs entrées

Pour le modèle à plusieurs entrées indépendantes comme le montre la Figure IV-3, l'écart type de sortie est composé comme [SAL-00], [WIK-02] :

$$\sigma_y = \sqrt{\sum_{i=1}^n \sigma_{yi}^2} \quad \text{Eq. IV-9}$$

où σ_{yi} est l'écart type de sortie pour chaque paramètre d'entrée x_i .

IV.1.1.2. Méthode de Monte-Carlo

La méthode de Monte-Carlo est la plus classique et la plus largement utilisée pour propager des incertitudes. Elle est basée sur le tirage aléatoire de variables d'entrée à partir de leur distribution et un grand nombre de calcul du modèle. Cette méthode permet d'obtenir des informations précises sur le type de distribution et ses différents moments comme ceux d'ordre 1 et 2 qui nous intéressent, espérance (μ) et variance σ^2 (et donc écart type σ). Cette technique, très simple à implémenter car elle ne nécessite que la capacité de calcul du modèle, est très coûteuse. En effet, plusieurs dizaines de milliers d'évaluations du modèle sont souvent nécessaires.

$$\mu_y = \frac{1}{N} \sum_{i=1}^N y_i \quad \text{Eq. IV-10}$$

$$\sigma_y = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \mu_y)^2} \quad \text{Eq. IV-11}$$

où N est le nombre de tirage aléatoire de variables d'entrée, y_i est la sortie correspondante à chaque tirage.

Dans cette technique, les variables aléatoires varient toutes en même temps. Ainsi le résultat obtenu tient compte en particulier des interactions entre variables dans le modèle, ce dont une approximation linéaire ne tient pas compte. Par contre, le fait de faire varier simultanément toutes les variables aléatoires ne permet pas de rendre compte de l'influence spécifique de chaque variable sur les sorties du modèle, ce que la méthode locale par dérivées partielles du premier ordre permet.

IV.1.2. Implémentation dans CADES

Nous avons implémenté les trois services d'analyse dans *CadesCalculator* :

- ✚ Service de calcul de sensibilité/incertitude locale basée sur le calcul des dérivées du premier ordre.

- ✚ Service de calcul de sensibilité/incertitude globale basé sur l'approximation linéaire avec la méthode de différence finie (la différence finie à droite a été choisie pour limiter le nombre d'évaluation du modèle).

- ✚ Service de calcul de sensibilité/incertitude globale avec la méthode Monte-Carlo. Comme cette méthode est très coûteuse en temps de calcul, et que les calculs des tirages sont indépendants, nous avons parallélisé ces calculs sur un ordinateur équipé de plusieurs processeurs (mais nous ne les avons pas distribués sur plusieurs ordinateurs).

Nombre de processeurs	2	3	4
Facteur de division en temps	1,8	2,6	3,1

Tableau IV-1. Facteurs moyens de division en temps pour le calcul parallèle sur un ordinateur de plusieurs processeurs [BAR-10]

Le Tableau IV-1 montre les facteurs de division en temps moyen, obtenus pour le calcul parallèle sur un ordinateur possédant plusieurs processeurs.

L'interface graphique des services de calcul de sensibilité/incertitude dans *CadesCalculator* est présentée dans l'annexe B.

IV.1.3. Application : analyse du nano commutateur magnétique

Le cahier des charges du nano commutateur magnétique a été présenté dans la section I.1.4.3 du premier chapitre. Son bon fonctionnement est dominé par la qualité du contact. Le contact est qualifié par sa résistance qui dépend de beaucoup de facteurs comme indiqués dans la section I.1.4.3 du premier chapitre. Dans cette thèse, nous ne considérons que les deux premiers facteurs : la force du contact et la longueur du contact. Dans cette section, nous allons appliquer les outils d'analyse de *CadesCalculator* pour étudier la sensibilité et/ou l'incertitude du modèle du nano commutateur magnétique (donc de la force du contact et de la longueur du contact) avec la variation (incertitude) des paramètres d'entrée du modèle.

a) Géométrie initiale

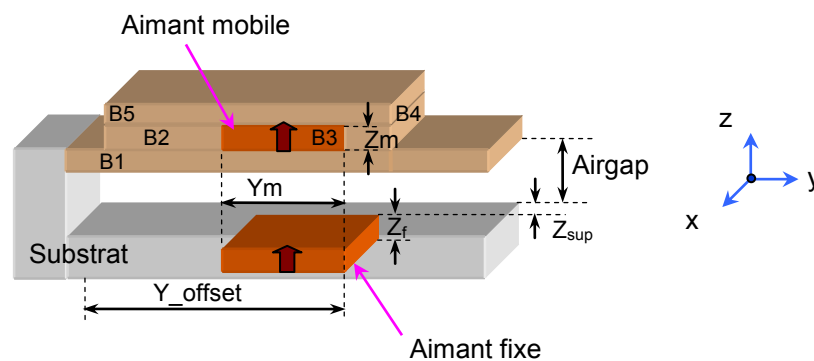


Figure IV-4. Géométrie initiale du nano commutateur magnétique

La Figure IV-4 montre la géométrie initiale du nano commutateur magnétique à analyser. La poutre est subdivisée en parallélépipèdes pour faciliter le paramétrage. Les paramètres géométriques et physiques sont détaillés dans l'annexe C.

Le Tableau IV-2 montre les résultats d'analyse avec *CadesCalculator*. Tous les écarts types des 19 entrées sont fixés à 5%

	Unité	Monte Carlo (Nbr=10 ⁵) nbrProcesseur=2	Approximation linéaire	
			dérivée 1 ^{ère}	différence finie à droite nbrProcesseur=2
Temps de calcul		2,5 [h]	4,7 [s]	2,3 [s]
Fcontact	10 ⁻⁸ N	3,280	3,279	3,279
σFcontact	10 ⁻⁸ N	0,537	0,536	0,536
σFcontact%	%	16,4	16,4	16,3
Lcontact	nm	139,9	143,4	143,4
σLcontact	nm	49,64	49,18	49,52
σLcontact%	%	35,5	34,3	34,5

Tableau IV-2. Ecarts types des contraintes Fcontact et Lcontact avec différentes méthodes (tous les écarts types des entrées étant à 5% de la géométrie initiale du nano commutateur magnétique)

Commentaires

✓ La méthode de propagation par Monte-Carlo est très coûteuse, mais elle donne des résultats précis considérés comme notre référence.

✓ Les approximations linéaires, utilisant la dérivée du premier ordre ou la différence finie à droite, donne de bons résultats en quelques secondes, comparées aux 2,5 heures de la méthode Monte-Carlo. Ces bons résultats, pour des incertitudes de 5%, montrent que notre modèle est relativement linéaire au voisinage de la géométrie considérée.

✓ 5% d'écart type sur les entrées engendrent ~16% d'écart type sur la force du contact et surtout ~35% d'écart type sur la longueur du contact. On peut conclure qu'avec cette géométrie initiale, le problème est trop sensible. Pour diminuer la sensibilité sur les contraintes, nous pouvons considérer deux voies :

- Chercher un dimensionnement réduisant cette sensibilité, mais nous ne pouvons espérer la réduire drastiquement.
- Chercher une structure différente pour garantir une plus faible sensibilité, et donc une meilleure fiabilité. C'est cette piste que nous avons suivie dans un premier temps.

b) Amélioration de la géométrie

Pour faciliter la déformation de la poutre pour créer le contact, nous avons proposé la géométrie illustrée sur la Figure IV-5, avec deux zones affaiblies de la poutre (parallélépipèdes B1 et B3). Les paramètres géométriques et physiques sont détaillés dans l'annexe C.

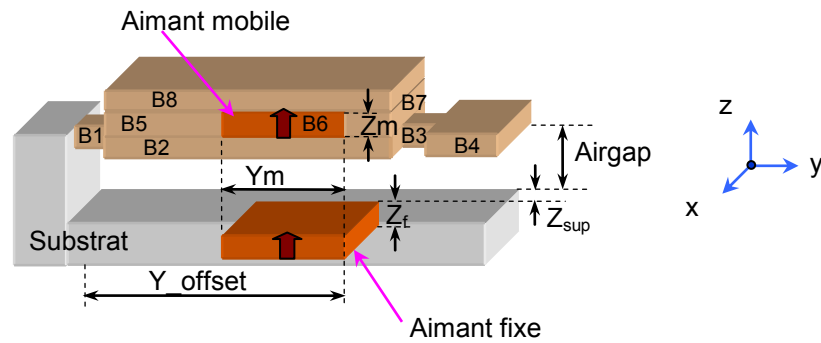


Figure IV-5. Géométrie améliorée du nano commutateur magnétique

Le Tableau IV-3 montre les résultats d'analyse avec *CadesCalculator*. Tous les écarts types des entrées sont fixés à 5%.

	unité	Monte-Carlo (Nbr=10 ⁵) nbrProcesseur=2	Approximation linéaire	
			dérivée 1 ^{ère}	différence finie à droite nbrProcesseur=2
Temps de calcul		2,4 [h]	4,8 [s]	2,3 [s]
Fcontact	10 ⁻⁸ N	3,967	3,956	3,956
σFcontact	10 ⁻⁸ N	0,562	0,559	0,555
σFcontact%	%	14,2	14,1	14,0
Lcontact	nm	352,40	353,40	353,40
σLcontact	nm	24,69	24,54	24,51
σLcontact%	%	7,01	6,95	6,94

Tableau IV-3. Ecarts types des contraintes Fcontact et Lcontact avec différentes méthodes (tous les écarts types des entrées étant à 5% de la géométrie améliorée du nano commutateur magnétique)

Commentaires

✓ En comparant avec la géométrie initiale, avec les mêmes dimensions des aimants, le nano commutateur assure la meilleur qualité de contact (la force de contact

est plus grande et la longueur de contact est plus importante). Ce qui justifie l'introduction des deux zones affaiblies créées par les parallélépipèdes B1 et B3.

✓ La sensibilité du modèle est bien réduite : 5% d'incertitude des entrées engendrent ~7% d'incertitude sur la longueur du contact et ~14% d'incertitude sur la force du contact.

✓ Cette configuration sera donc utilisée, pour l'étape suivante visant à faire un dimensionnement optimal du nano commutateur magnétique.

IV.2. Optimisation sous contraintes du nano commutateur magnétique

IV.2.1. Cahier des charges

Le Tableau IV-4 montre le cahier des charges pour l'optimisation du nano commutateur magnétique. Le dimensionnement optimal consiste à déterminer les dimensions et les positions des aimants (mobile et fixe), pour minimiser le volume de l'aimant en assurant la qualité de contact.

Objectif	V_magnet	Volume des aimants	A minimiser
Contraintes	Lcontact	Longueur de contact	≥ 300 nm
	Fcontact	Force de contact	$\geq 1E-8$ N
Paramètres à optimiser	Ym	Longueur des aimants (fixe et mobile)	[100 : 1000] nm
	Zf	Hauteur de l'aimant fixe	[100 : 1000] nm
	Zm	Hauteur de l'aimant mobile	[100 : 1000] nm
	Y_offset	Position des aimants sur la poutre	[2500 : 3500] nm

Tableau IV-4. Cahier des charges du nano commutateur magnétique

Nous allons faire l'étude des deux configurations suivantes selon la direction d'aimantation des aimants, comme illustré sur la Figure IV-6.

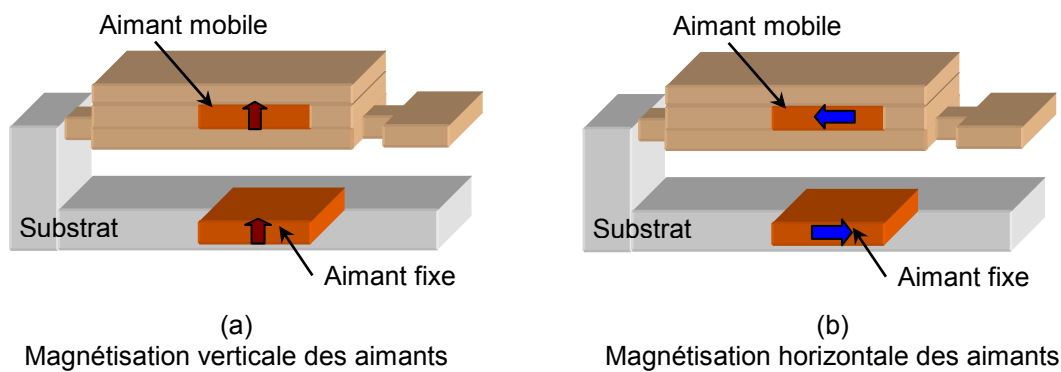


Figure IV-6. Deux configurations du nano commutateur magnétique à optimiser

IV.2.2. Stratégies d'optimisation

Nous avons, dans le chapitre I, étudié l'intérêt de disposer des gradients pour l'optimisation. Ainsi, une première stratégie utilisera un algorithme stochastique (sans gradients), puis une seconde stratégie indirecte utilisera la surface de réponse adaptative, et la troisième stratégie utilisera un algorithme déterministe exploitant le Jacobien du modèle. Les méthodes sont décrites dans un premier temps, les résultats sont analysés par la suite.

IV.2.2.1. Optimisation directe par algorithme stochastique

Nous avons utilisé un algorithme PSO¹ [KEN-95] pour optimiser le nano commutateur magnétique avec 50 individus et 400 générations menant à 17000 évaluations du modèle pendant environ 50 minutes.

Cette stratégie d'optimisation directe d'un modèle de calcul est relativement lourde, surtout durant les phases de conception durant lesquelles le nombre d'optimisations est important. En effet, la formulation du cahier des charges peut évoluer en fonction de l'algorithme, de la prise en compte des contraintes, le modèle peut être adapté, le cahier des charges relâché, etc. Pour ces raisons, des stratégies indirectes, comme celle que nous allons mettre en œuvre maintenant, doivent être utilisés. Ceci est d'autant plus important que le modèle de calcul est coûteux (modèle éléments finis par exemple).

¹PSO : Particle Swarm Optimization

IV.2.2.2. Optimisation indirecte par surface de réponse adaptative

Une surface de réponse est une approximation de la fonction objectif et les fonctions de contrainte du modèle. Elle est construite à partir des résultats calculés pour des expériences bien choisies, puis par application d'un algorithme d'interpolation. Bien que ce soit une approximation, elle fournit une connaissance globale du comportement du modèle. Les représentations mathématiques de surface de réponse sont [COU-02] : les polynôme du 1^{er} degré, les polynômes du second degré, les fonctions radiales ou sur les éléments diffus, dans lesquelles les polynômes du second degré sont couramment utilisés.

Comme la surface de réponse est une approximation, l'optimisation sur tout l'espace de recherche conduit à une erreur sur les résultats. Pour améliorer et raffiner la technique de surface de réponse, une méthodologie de surface de réponse adaptative a été implémenté dans FGOT¹ (« *Adaptive Response Surface* » en anglais [WAN-03]), où l'espace de recherche est réduit de façon itérative et centré autour de la solution optimisée sur chaque itération.

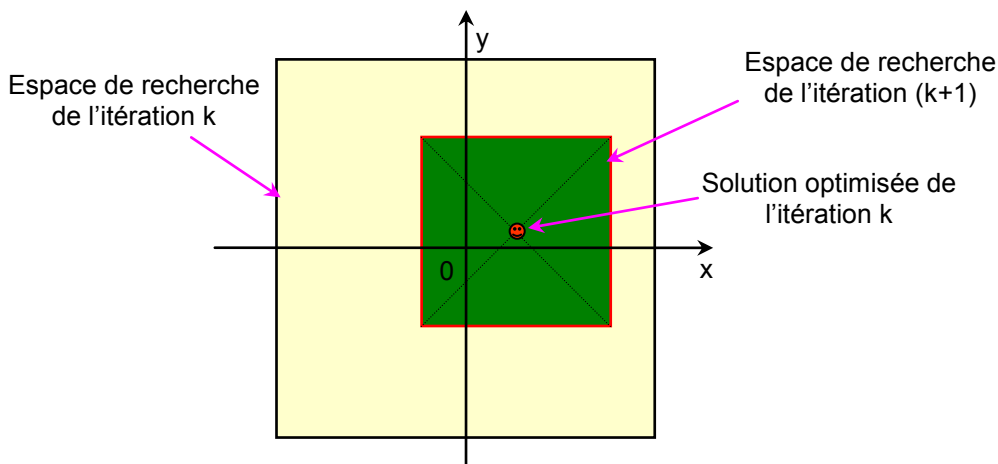


Figure IV-7. Exemple en deux dimensions de la réduction de l'espace de recherche (divisée par 2) à chaque itération

A chaque itération, on optimise le problème sous contraintes avec les fonctions d'objectifs et les fonctions de contrainte qui sont remplacées par des surfaces de réponse construites sur le domaine actuel. La solution optimisée à cette itération est utilisée pour définir un nouveau domaine pour la prochaine itération comme illustré un exemple sur la Figure IV-7.

¹ FGOT : *Featuring a Genuine Optimization Tool* [COU]

Dans notre étude sur le nano commutateur magnétique, nous avons choisi 10 itérations avec les polynômes du second degré. La réduction de l'espace de recherche en 5 premières itérations lors de l'optimisation du cas (a) Figure IV-6 est montrée dans le Tableau IV-5.

	1 ^{ère} itération	2 ^e itération	3 ^e itération	4 ^e itération	5 ^e itération
Ym [nm]	[100 : 1000]	[325 : 775]	[212,5 : 437,5]	[212,5 : 325]	[296,9 : 353,1]
Zf [nm]	[100 : 1000]	[100 : 550]	[100 : 325]	[212,5 : 325]	[184,4 : 240,6]
Zm [nm]	[100 : 1000]	[100 : 550]	[100 : 325]	[212,5 : 325]	[184,4 : 240,6]
Y_offset [nm]	[2500 : 3500]	[3000 : 3500]	[3250 : 3500]	[3375 : 3500]	[3437,5 : 3500]

Tableau IV-5. Réduction de l'espace de recherche lors de l'optimisation du nano commutateur magnétique du cas (a) de Figure IV-6

IV.2.2.3. Optimisation par un algorithme déterministe exploitant le Jacobien

L'algorithme déterministe utilisé (SQP¹) exploite le calcul des gradients. Cet algorithme converge rapidement (une dizaine d'itération durant une vingtaine de secondes avec la précision 10^{-5} pour le nano commutateur magnétique) et a la capacité de satisfaire un cahier des charges fortement contraint (quelques centaines de paramètres et de contraintes). Cependant, lorsque le problème possède plusieurs optimums, il peut converger vers une solution locale qui dépend des valeurs initiales.

Pour utiliser l'algorithme SQP, la tâche la plus difficile est le calcul précis des dérivées. Grâce aux techniques de Dérivation Automatique de JAP, cette tâche est devenue transparente pour le concepteur. Dans la section III.5.2.4. , la composition du Jacobien du modèle du nano commutateur a été présentée grâce à JAP.

¹ SQP : Sequential Quadratic Programming

IV.2.3. Résultats d'optimisation

	Valeurs initiales	Valeurs optimisées		
		PSO	Surface de réponse adaptative (10 itérations)	SQP (13 itérations)
Modèle évaluations		~17000	307	13
Temps d'optimisation		50 [min]	142,5 [s]	24,4 [s]
Ym [nm]	500	328,01	330,31	330,01
Zf [nm]	500	207,91	208,79	208,95
Zm [nm]	500	211,96	208,03	208,24
Y_offset [nm]	3000	3500	3500	3500
Lcontact [nm] (≥ 300)	353,4	300,00	300,00	300,00
Fcontact [10E-8N] (≥ 1)	3,956	2,116	2,115	2,115
V_magnet [1E-20 m ³] (à minimiser)	25,0	6,886	6,884	6,884

Tableau IV-6. Résultats d'optimisation du nano commutateur magnétique cas d'aimantation verticale des aimants (Figure IV-6 –a)

	Valeurs initiales	Valeurs optimisées		
		PSO	Surface de réponse adaptative (10 itérations)	SQP (9 itérations)
Modèle évaluations		~17000	300	9
Temps d'optimisation		40 [min]	120,8 [s]	15,5 [s]
Ym [nm]	500	425,05	424,88	425,19
Zf [nm]	500	436,49	435,74	434,82
Zm [nm]	500	432,91	433,98	434,25
Y_offset [nm]	3000	3500	3500	3500
Lcontact [nm] (≥ 300)	287,72	300,00	300,00	300,00
Fcontact [10E-8N] (≥ 1)	1,882	2,117	2,117	2,117
V_magnet [1E-20 m ³] (à minimiser)	25,0	18,477	18,476	18,476

Tableau IV-7. Résultats d'optimisation du nano commutateur magnétique cas d'aimantation horizontale des aimants (Figure IV-6 –b)

Commentaires

✓ Tout d'abord, on peut constater que les trois stratégies trouvent presque les mêmes résultats, ce qui peut être expliqué par l'uni modalité du problème.

✓ Avec l'usage des dérivées comme guide vers l'optimum, l'algorithme SQP converge le plus vite après quelques itérations (13 itérations pour le cas d'aimantation verticale, 9 itérations pour le cas d'aimantation horizontale) en un temps considérablement réduit (120 fois plus vite par rapport à PSO et 6 fois plus vite par rapport à surface de réponse adaptative).

✓ L'algorithme de surface de réponse adaptative est assez efficace. Cet algorithme est applicable sur les modèles sans Jacobien. Il est aussi efficace pour des problèmes dans lesquels les fonctions d'objectifs et/ou contraintes oscillent [NGU-10] [PAR-05].

✓ Concernant la conception du nano commutateur magnétique, la configuration d'aimantation verticale est la meilleure par rapport à celle d'aimantation horizontale (volume d'aimant – fonction objectif est plus petite 6,884 par rapport à 18,476).

IV.2.4. Sensibilité de la solution optimale

L'optimisation telle que nous l'avons vue dans la section précédente permet de trouver une solution optimale nominale. C'est-à-dire que les performances estimées du système répondent au mieux aux valeurs définies dans le cahier des charges. Ces résultats ne tiennent pas compte de l'impact des facteurs incertains du modèle. Pour évaluer l'impact des incertitudes des entrées sur les contraintes, nous avons utilisé les outils d'analyse de sensibilité/incertitude de *CadesCalculator*. Avec tous les écarts types des entrées fixés à 5%, nous obtenons les incertitudes des contraintes (F_{contact} et L_{contact}) comme illustrées dans le Tableau IV-8.

	unité	Monte-Carlo (Nbr=10 ⁵) nbrProcesseur=2	Approximation linéaire	
			dérivée 1 ^{ère}	différence finie à droite nbrProcesseur=2
Temps de calcul		2,5 [h]	4,6 [s]	2,6 [s]
F_{contact}	10 ⁻⁸ N	2,032	2,115	2,115
$\sigma F_{\text{contact}}$	10 ⁻⁸ N	0,253	0,225	0,264
$\sigma F_{\text{contact}}\%$	%	12,40	10,65	12,50
L_{contact}	nm	296,40	300	300
$\sigma L_{\text{contact}}$	nm	31,24	29,27	29,81
$\sigma L_{\text{contact}}\%$	%	10,5	9,76	9,94

Tableau IV-8. Ecarts types des contraintes F_{contact} et L_{contact} avec tous les écarts types fixés à 5% du nano commutateur magnétique optimisé (aimantation verticale des aimants)

Commentaires

✓ 5% d'écart type des entrées engendrent ~10% d'écart type sur la longueur du contact, et ~12% d'écart type sur la force de contact. C'est-à-dire que 5% d'incertitude des entrées engendrent 12% d'incertitude des contraintes. Le nano commutateur magnétique risque de ne pas fonctionner (la résistance de contact n'étant pas assurée). Pour assurer le bon fonctionnement (fiable) du nano commutateur magnétique, il faut tenir compte de ces incertitudes dans l'optimisation. Ce type d'optimisation est appelé l'optimisation sous contraintes de fiabilité et va être traitée dans la section suivante de ce chapitre.

IV.3. Optimisation sous contraintes de fiabilité

IV.3.1. Contrôle de qualité : Six sigma (6σ)

De nombreux processus de fabrication obéissent à une loi normale. Les mesures sur les produits fabriqués permettent d'établir les caractéristiques de la courbe de distribution de Gauss, et de connaître la valeur moyenne et l'écart type. Ce qui est important pour maîtriser un processus de fabrication est que les valeurs mesurées (du produit) restent dans l'intervalle de tolérance fixée par le cahier des charges, c'est-à-dire entre la limite inférieure de spécification - LSL¹, et la limite supérieure de spécification - USL². Les produits qui sortent d'un processus de fabrication doivent être à l'intérieur des limites spécifiées avec une fiabilité fixée par le cahier des charges.

Si on réduit l'incertitude (variabilité) des produits du processus, on réduit le risque de voir le produit rejeté pour non respect du cahier des charges. Statistiquement, plus l'incertitude est petite plus la fiabilité est assurée. La Figure IV-8 illustre la distribution de Gauss avec la valeur moyenne $\mu=0$ et l'écart type $\sigma=1$.

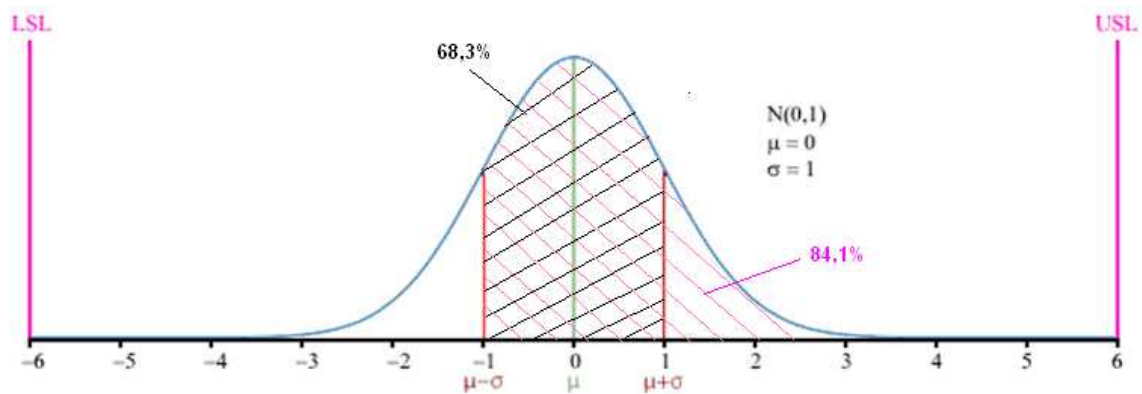


Figure IV-8. Distribution de Gauss avec l'incertitude $k.\sigma$

Le Tableau IV-9 montre le taux de fiabilité correspondant à l'incertitude $k.\sigma$ ($k \in \mathbb{N}$). Ce taux de fiabilité est une intégration de la fonction de densité de probabilité de Gauss $p(x)$.

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad \text{Eq. IV-12}$$

¹ LSL : Low Specification Limit

² USL : Upper Specification Limit

$$P(\mu - k.\sigma \leq X \leq \mu + k.\sigma) = \int_{\mu - k.\sigma}^{\mu + k.\sigma} p(x)dx \quad \text{Eq. IV-13}$$

$$P(X \geq \mu - k.\sigma) = P(X \leq \mu + k.\sigma) = \int_{\mu - k.\sigma}^{+\infty} p(x)dx = \int_{-\infty}^{\mu + k.\sigma} p(x)dx \quad \text{Eq. IV-14}$$

k.σ	σ	2σ	3σ	4σ	5σ	6σ
Fiabilité symétrique (dans l'intervalle μ±k.σ)	68%	95,4%	99,73%	99,994%	99,99994%	99,9999998%
Fiabilité d'un côté (≥μ-k.σ ou ≤μ+k.σ)	84%	97,7%	99,87%	99,997%	99,99997%	99,9999999%

Tableau IV-9. Fiabilité correspondant à l'incertitude de k.σ

Pour bien expliquer le Tableau IV-9, on prend l'exemple d'un NEMS dont la longueur nominale est L. La tolérance acceptée est ± ΔL, c'est-à-dire que le NEMS est utilisable si la longueur est comprise entre L-ΔL et L+ΔL (fiabilité symétrique, ou contrainte d'encadrement). Le processus de fabrication, lui, produit des pièces avec une incertitude sur cette longueur d'écart type σ :

✓ Si le processus de fabrication est réglé afin que l'on ait ΔL = 6σ (soit σ = ΔL/6), on aura un taux de rebut de 0,0000002%, soit 2 NEMS au rebut pour un milliard de produits. Ce processus est donc presque parfait. Inspirer de cette caractéristique, 6σ (six sigma) est devenue une méthode structurée de management visant à une amélioration de la qualité et de l'efficacité des processus comme : des procédés industriels, processus administratifs, logistiques, commerciaux etc. [WIK-3].

✓ En pratique la limite ΔL=6σ est difficilement atteignable, mais des valeurs plus faibles sont utilisées telles que 3σ ou 4σ (soit taux de rebut de 0,27% ou 0,006%). Cependant certaines applications exigeantes comme dans l'aéronautique peuvent demander un objectif de 20σ [PEY-03].

✓ Si la tolérance acceptée est ≥L-ΔL (fiabilité d'un côté, ou contrainte de supériorité ou d'infériorité) et ΔL=3σ, on a un taux de fiabilité de 99,87% soit un taux de rebut de 0,13%.

En appliquant cette technique au nano commutateur magnétique avec la limite 3σ , on peut expliquer statistiquement les résultats du Tableau IV-8 de la manière suivante :

✓ On considère que le processus de fabrication du nano commutateur magnétique respecte la limite 3σ soit $\Delta=3\sigma=3*5\%=15\%$ d'incertitude. On a donc l'écart type de tous les paramètres d'entrée $\sigma=5\%$.

✓ Avec la limite 3σ sur la qualité du contact du nano commutateur magnétique, les résultats d'analyse de sensibilité/incertitude donnent :

- écart type de longueur de contact $\sigma_{L_{\text{contact}}}\approx 10\%$ soit $\Delta L_{\text{contact}}\approx 3*10\% = 30\%$, $\Delta L_{\text{contact}}\approx 90$ [nm].

- écart type de force de contact $\sigma_{F_{\text{contact}}}\approx 12\%$ soit $\Delta F_{\text{contact}}\approx 3*12\% = 36\%$, $\Delta F_{\text{contact}}\approx 0,75$ [10^{-8} N].

On a les probabilités de F_{contact} et L_{contact} montrées dans le Tableau IV-10.

	unité	$X\geq\mu$	$\mu-3\sigma\leq X\leq\mu+3\sigma$	$X\geq\mu-3\sigma$
L_{contact}	[nm]	≥ 300	300 ± 90	$\geq 300-90$
F_{contact}	[10^{-8} N]	$\geq 2,12$	$2,12\pm 0,75$	$\geq 2,12-0,75$
Probabilité	[%]	50	99,73	99,87

Tableau IV-10. Probabilités de F_{contact} et L_{contact}

✓ On en déduit que, si la configuration optimisée dans le Tableau IV-8 est fabriquée, seulement 50% des nano commutateurs magnétiques satisfont la qualité du contact (cahier des charges du Tableau IV-4), et donc que 50% partent au rebut. Pour améliorer ce taux (jusqu'à 3σ par exemple), il faut tenir compte ces incertitudes dans les contraintes de l'optimisation.

IV.3.2. Méthode découplée de résolution du problème d'optimisation sous contraintes de fiabilité

L'optimisation sous contraintes de fiabilité (RBDO : Reliability-Based Design Optimization) est l'optimisation en tenant compte des incertitudes du modèle. La RBDO cherche à déterminer une conception à coût minimal, tout en assurant une performance avec fiabilité demandée. La formulation du problème RBDO a été présentée dans l'équation Eq. I-6 de la section I.3.3.2. du premier chapitre. Pour plus de détails sur la RBDO, le lecteur peut consulter [TSO-08].

La solution du problème RBDO n'est pas facile, même pour des modèles simples, car il faut résoudre en même temps le problème d'optimisation et le problème de contraintes de fiabilité. Une première approche consisterait à utiliser une boucle d'optimisation sous contraintes dans laquelle on viendrait évaluer les contraintes probabilistes par la méthode de Monte-Carlo. Cette approche est très simple et donne des résultats précis, mais elle n'est pas applicable en pratique en raison de temps de calcul trop long. D'autres méthodes d'approximation calculant l'indice de fiabilité peuvent être utilisées telles que méthode de fiabilité du premier ou du second ordre (First/Second Order Reliability Method -FORM/SORM) [HAL-00], [SAL-00].

Des stratégies alternatives ont été proposées dans la littérature RBDO, voir [TBO-08] pour une revue des méthodes existantes. Dans la section I.3.3.2. du premier chapitre, nous avons présenté brièvement les trois principales méthodes citées dans [TBO-08] : l'approche à deux niveaux, l'approche à un niveau, et l'approche découplée. Nous avons choisi l'approche découplée pour l'outil d'optimisation sous contraintes de fiabilité que nous avons implémenté dans CADES.

L'approche découplée transforme la contrainte de fiabilité en contrainte déterministe approximée. Le problème RBDO peut être formulé comme :

$$\left\{ \begin{array}{ll} \text{- minimiser} & F(x) \\ \text{avec} & x = \{x_1, x_2, \dots, x_n\} \in \mathfrak{R}^n \\ \text{et} & x_i^{\min} \leq x_i \leq x_i^{\max} \quad i = 1, \dots, n \\ \text{- en respectant} & H_j(x) + k_\sigma \cdot \sigma_{H_j(x)} \leq 0 \quad j = 1, \dots, m \end{array} \right. \quad \text{Eq. IV-15}$$

où $\sigma_{H_j(x)}$ sont des écarts types des fonctions de contrainte $H_j(x)$, k_σ est le coefficient correspondant au niveau de fiabilité demandé ($k_\sigma=3$ pour 99,87%, $k_\sigma=4$ pour 99,997%, etc.).

L'avantage de cette approche est qu'elle ne demande pas d'analyse de fiabilité séparément. L'analyse de fiabilité est déjà couplée intrinsèquement avec le problème d'optimisation. Le problème RBDO devient donc un problème d'optimisation standard. Mais la limite de cette méthode est l'approximation linéaire des écarts types $\sigma_{H_j(x)}$ (par la dérivée de premier ordre ou par différence finie).

Afin d'utiliser l'algorithme d'optimisation déterministe tel que SQP pour résoudre le problème RBDO, le calcul des dérivées des $\sigma_{H_j(x)}$ est demandé :

✓ Si $\sigma_{H_j(x)}$ sont approximés par les dérivées de premier ordre, pour calculer $\frac{\partial \sigma_{H_j(x)}}{\partial x}$, il faut calculer les dérivées de second ordre $\frac{\partial^2 H_j(x)}{\partial x^2}$. Pour des raisons de coût, ce Hessien (Jacobien du Jacobien) n'est pas calculé dans CADES. Une solution serait l'utilisation de différences finies sur les dérivées premières qui seraient alors exploitées par l'optimisation. Malheureusement, une telle approche souffre de problème de convergence de l'algorithme d'optimisation.

✓ Si $\sigma_{H_j(x)}$ sont approximés par différences finies, les dérivées $\frac{\partial \sigma_{H_j(x)}}{\partial x}$ sont calculées facilement par les dérivées de premier ordre $\frac{\partial H_j(x)}{\partial x}$ dans CADES. C'est cette solution qui a été retenue, et qui est détaillée dans la section suivante.

IV.3.3. CadesDeviationGenerator

IV.3.3.1. Principe de CadesDeviationGenerator

CadesDeviationGenerator est un outil du framework CADES pour faire l'optimisation sous contraintes de fiabilité. Il propage des écarts types des entrées en écarts types sur les sorties.

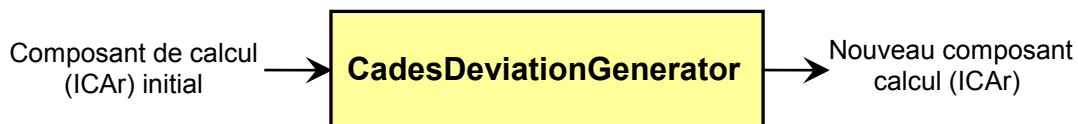


Figure IV-9. Entrée/sortie du *CadesDeviationGenerator*

CadesDeviationGenerator génère le nouveau composant de calcul (ICAr¹) à partir du composant de calcul initial comme illustré sur la Figure IV-9.

Dans le nouveau composant de calcul, *CadesDeviationGenerator* encapsule le composant de calcul initial, et lui ajoute des entrées et des sorties nécessaires à la propagation des incertitudes comme illustré sur la Figure IV-10.

¹ ICAr : Interfaces for Component Architecture – Le nom de la norme du composant logiciel de CADES

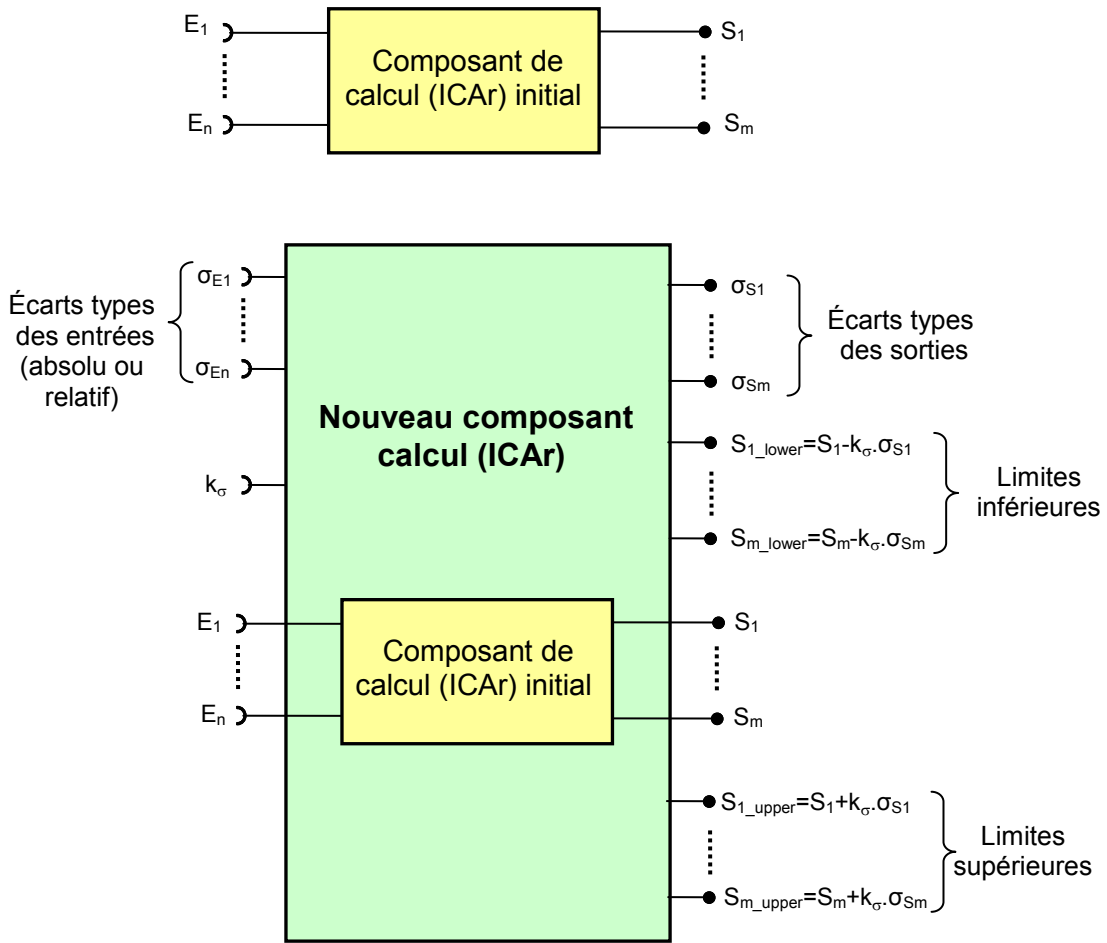


Figure IV-10. Entrées/sorties du nouveau composant de calcul (ICAr) généré par *CadesDeviationGenerator* par rapport à ceux du composant de calcul (ICAr) initial

✓ En entrée : il ajoute l'écart type pour chaque entrée (σ_{E_i} , $i \in [1, n]$) et le coefficient k_σ correspondant au niveau de fiabilité demandé ($k_\sigma=3$ pour 99,87%, $k_\sigma=4$ pour 99,997%, etc.).

✓ En sortie : il ajoute l'écart type pour chaque sortie (σ_{S_j} , $j \in [1, m]$) ainsi que des limites inférieures ($S_{j_lower} = S_j - k_\sigma \cdot \sigma_{S_j}$) et des limites supérieures ($S_{j_upper} = S_j + k_\sigma \cdot \sigma_{S_j}$). Les contraintes (voir Eq. IV-16) dans l'optimisation standard sont remplacées par celles de Eq. IV-17 dans l'optimisation sous contraintes de fiabilité.

$$S_{j \min} \leq S_j \leq S_{j \max} \quad \text{Eq. IV-16}$$

$$\begin{aligned} S_{j \min} &\leq S_{j_lower} = S_j - k_\sigma \cdot \sigma_{S_j} \\ S_{j \max} &\geq S_{j_upper} = S_j + k_\sigma \cdot \sigma_{S_j} \end{aligned} \quad \text{Eq. IV-17}$$

Des écarts types des sorties sont calculés par Eq. IV-18 :

$$\sigma_{S_j} = \sqrt{\sum_{k=1}^n \sigma_{S_{j-k}}^2} \quad \text{Eq. IV-18}$$

où $\sigma_{S_{j-k}}$ est l'écart type de sortie S_j pour chaque paramètre d'entrée E_k ($k \in [1, n]$)

$\sigma_{S_{j-k}}$ sont calculés par différence finie à droite :

$$\sigma_{S_{j-k}} = S_j(E_1, \dots, (E_k + \sigma_{E_k}), \dots, E_n) - S_j(E_1, \dots, E_n) \quad \text{Eq. IV-19}$$

où $S_j(E_1, \dots, (E_k + \sigma_{E_k}), \dots, E_n)$ est la sortie S_j calculée par le composant de calcul initial avec les entrées $E_1, \dots, (E_k + \sigma_{E_k}), \dots, E_n$.

et $S_j(E_1, \dots, E_n)$ est la sortie S_j calculée par le composant de calcul initial avec les entrées E_1, \dots, E_n .

Comme les calculs de Eq. IV-19 sont indépendants, la parallélisation sur un ordinateur multi cœurs a été implémenté dans *CadesDeviationGenerator*.

IV.3.3.2. Composition du Jacobien dans CadesDeviationGenerator

Nous présentons ici comment le Jacobien du modèle global (avec propagation d'incertitude) est constitué à partir du Jacobien du modèle de calcul initial. On ne s'intéresse pas aux dérivées par rapport aux écarts types des entrées (σ_{E_k}) et par rapport au coefficient k_σ . La composition du Jacobien de *CadesDeviationGenerator* consiste donc à calculer les dérivées des écarts types des sorties en fonctions des entrées $\frac{\partial(\sigma_{S_j})}{\partial E_i}$ ($i \in [1, n], j \in [1, m]$). Les dérivées de S_{j_upper} et S_{j_lower} sont calculées par :

$$\frac{\partial S_{j_upper}}{\partial E_i} = \frac{\partial S_j}{\partial E_i} + k_\sigma \cdot \frac{\partial(\sigma_{S_j})}{\partial E_i} \quad \text{Eq. IV-20}$$

$$\frac{\partial S_{j_lower}}{\partial E_i} = \frac{\partial S_j}{\partial E_i} - k_\sigma \cdot \frac{\partial(\sigma_{S_j})}{\partial E_i} \quad \text{Eq. IV-21}$$

où $\frac{\partial S_j}{\partial E_i} = J_0[j][i]$, J_0 est la matrice jacobienne (de taille $m \times n$) du composant de calcul initial calculée avec les entrées E_1, \dots, E_n .

Nous allons détailler la composition pour $\frac{\partial(\sigma_{S_j})}{\partial E_i}$:

On a :

$$\begin{aligned} \frac{\partial(\sigma_{S_j})}{\partial E_i} &= \frac{\partial(\sqrt{\sum_{k=1}^n \sigma_{S_{j-k}}^2})}{\partial E_i} \quad (\text{par Eq. IV-18}) \\ &= \frac{1}{2 \cdot \sigma_{S_j}} \cdot \sum_{k=1}^n (2 \cdot \sigma_{S_{j-k}} \cdot \frac{\partial(\sigma_{S_{j-k}})}{\partial E_i}) \quad (\text{car } \frac{\partial(\sqrt{u})}{\partial x} = \frac{1}{2 \cdot \sqrt{u}} \cdot \frac{\partial u}{\partial x}) \\ &= \frac{1}{\sigma_{S_j}} \cdot \sum_{k=1}^n (\sigma_{S_{j-k}} \cdot \frac{\partial(\sigma_{S_{j-k}})}{\partial E_i}) \end{aligned}$$

Par Eq. IV-19 on a :

$$\frac{\partial(\sigma_{S_{j-k}})}{\partial E_i} = \frac{\partial S_j(E_1, \dots, (E_k + \sigma_{E_k}), \dots, E_n)}{\partial E_i} - \frac{\partial S_j(E_1, \dots, E_n)}{\partial E_i} =$$

$$J_k[j][i] \cdot (1 + \frac{\partial(\sigma_{E_k})}{\partial E_i}) - J_0[j][i]$$

où :

✓ J_0 est la matrice jacobienne du composant de calcul initial calculée avec les entrées E_1, \dots, E_n .

✓ J_k est la matrice jacobienne du composant de calcul initial calculée avec des entrées $E_1, \dots, (E_k + \sigma_{E_k}), \dots, E_n$.

$$\checkmark \quad \frac{\partial(\sigma_{E_k})}{\partial E_i} = \begin{cases} \sigma_{E_k \%} & \text{si } k = i \quad \text{avec } \sigma_{E_k \%} \text{ est l'écart type relatif [\%]} \\ 0 & \text{si non} \end{cases}$$

On en déduit donc :

$$\frac{\partial(\sigma_{S_j})}{\partial E_i} = \frac{1}{\sqrt{\sum_{k=1}^n \sigma_{S_{j-k}}^2}} \cdot \left(-\sum_{k=1}^n (\sigma_{S_{j-k}}) \cdot J_0[j][i] + \sum_{k=1}^n (\sigma_{S_{j-k}} \cdot J_k[j][i] \cdot (1 + \frac{\partial(\sigma_{E_k})}{\partial E_i})) \right) \quad \text{Eq. IV-22}$$

où $\sigma_{S_{j-k}}$ sont calculés par Eq. IV-19.

Les calculs des matrices jacobiennes J_0 et J_k ($k \in [1, n]$) sont indépendants, et donc la parallélisation a été aussi implémentée dans *CadesDeviationGenerator* pour gagner en temps de calcul.

IV.3.4. Application pour le nano commutateur magnétique

IV.3.4.1. Résultats d'optimisation

Comme le composant de calcul est généré par *CadesDeviationGenerator* qui fournit le Jacobien, nous pouvons utiliser l'algorithme déterministe SQP pour faire l'optimisation sous contraintes de fiabilité du nano commutateur magnétique. Le Tableau IV-11 montre les résultats d'optimisation avec trois cas de k_σ correspondant à trois niveaux de fiabilité différents.

	Valeurs initiales	Valeurs optimisées sans fiabilité	Valeurs optimisées avec fiabilité		
			$k_\sigma=2$	$k_\sigma=3$	$k_\sigma=4$
Fiabilité		50%	97,7%	99,87%	99,997%
Nombre d'itérations		13	13	16	14
Temps d'optimisation		24,4[s]	219[s]	280[s]	262[s]
Ym [nm]	500	330,01	529,80	690,73	927,34
Zf [nm]	500	208,95	276,01	314,11	359,40
Zm [nm]	500	208,24	274,64	312,24	356,75
Y_offset [nm]	3500	3500	3500	3500	3500
Lcontact [nm]		300,00	349,25	369,15	387,20
σ Lcontact [nm]		29,81	24,63	23,05	21,80
Lcontact_lower [nm] (≥ 300)			300,00	300,00	300,00
Fcontact [10E-8N]		2,115	3,74	4,97	6,70
σ Fcontact [10E-8N]		0,264	0,437	0,567	0,750
Fcontact_lower [10E-8N] (≥ 1)			2,87	3,27	3,71
V_magnet [1E-20 m ³] (à minimiser)	25,0	6,884	14,59	21,63	33,21

Tableau IV-11. Résultats d'optimisation sous contraintes de fiabilité du nano commutateur magnétique avec $k_\sigma=2, 3, 4$

Commentaires

✓ Dans l'optimisation sans fiabilité, la contrainte de longueur du contact atteint sa limite ($L_{contact}=300\text{nm}$). Tandis que dans l'optimisation avec fiabilité, cette contrainte est remplacée par $L_{contact_lower}=L_{contact}-k_{\sigma}\cdot\sigma_{L_contact}$.

✓ Plus k_{σ} est grand plus la fiabilité est grande et plus la fonction d'objectif (V_{magnet}) est grande. C'est-à-dire qu'il faut payer pour assurer le niveau de fiabilité.

✓ Quant au temps d'optimisation : l'optimisation avec fiabilité est plus longue que l'optimisation normale. Ce facteur varie en fonction du nombre de paramètres d'entrée incertains du modèle, et aussi du nombre de processeur de l'ordinateur. Dans notre cas, on peut expliquer le facteur 10 par $\sim 19/2$ avec 19 paramètres d'entrée incertains et 2 processeurs du PC utilisé.

IV.3.4.2. Vérification des résultats par Monte-Carlo

Au cours de l'optimisation avec fiabilité sur le composant de calcul généré par *CadesDeviationGenerator*, les incertitudes sont propagées par la méthode des différences finies (DF) à droite se basant sur l'hypothèse de linéarité du modèle autour du point de calcul. Afin de tester la validité de cette hypothèse, on peut vérifier par la méthode de Monte-Carlo (MC) en utilisant le *CadesCalculator*. Les résultats sont dans le Tableau IV-12.

	$k_\sigma=2$		$k_\sigma=3$		$k_\sigma=4$	
	MC Nbr= 10^5	DF à droite	MC Nbr= 10^5	DF à droite	MC Nbr= 10^5	DF à droite
Lcontact [nm]	347,6	349,25	367,70	369,15	386,00	387,20
σ Lcontact [nm]	24,94	24,63	23,20	23,05	22,01	21,80
Lcontact_lower [nm] (>=300)	297,70	300,00	298,10	300,00	297,96	300,00
Fcontact [10E-8N]	3,60	3,74	4,79	4,97	6,45	6,70
σ Fcontact [10E-8N]	0,418	0,437	0,548	0,567	0,719	0,75
Fcontact_lower [10E-8N] (>=1)	2,76	2,87	3,15	3,27	3,57	3,71

Tableau IV-12. Vérification des résultats d'optimisation sous contraintes de fiabilité du nano commutateur magnétique ($k_\sigma=2, 3, 4$) par la méthode de Monte-Carlo

Commentaires

✓ Les valeurs moyennes de Lcontact et de Fcontact sont un peu différentes des valeurs calculées aux points optimisées. Cela provient du fait que dans l'approximation linéaire, la valeur moyenne des sorties du modèle correspond au simple calcul du modèle pour les valeurs moyennes des entrées.

✓ Les écarts types calculés par Monte-Carlo sont assez proches de ceux calculés par différence finie (DF) à droite.

✓ A cause de ces différences, les contraintes de *Lcontact_lower* ne sont pas atteintes. C'est-à-dire que les fiabilités obtenues sont différents de celles désirées (97,7% pour $k_\sigma=2$ par exemple). Avec μ et σ calculés par Monte-Carlo, on peut évaluer les fiabilités effectives par Eq. IV-14, ces fiabilités sont montrées dans le Tableau IV-13.

	$k_\sigma=2$	$k_\sigma=3$	$k_\sigma=4$
Fiabilités désirées	97,7%	99,87%	99,997%
Fiabilités obtenues (μ et σ calculés par Monte-Carlo)	97,2%	99,82%	99,995%

Tableau IV-13. Fiabilités désirées/obtenues du nano commutateur magnétique après l'optimisation sous contraintes de fiabilité avec $k_\sigma=2, 3, 4$

IV.4. Conclusions

Dans ce chapitre, les méthodes et les outils d'optimisation sous contraintes de fiabilité ont été présentés et appliqués pour le nano commutateur magnétique. Tout d'abord, nous avons présenté les méthodes d'analyse de sensibilité et d'analyse d'incertitude et leur implémentation dans le *CadesCalculator*. Ensuite, nous avons présenté les différentes stratégies d'optimisation du nano commutateur magnétique pour montrer l'intérêt de l'algorithme déterministe exploitant le Jacobien. Afin de tenir compte de l'incertitude dans l'optimisation, nous avons proposé une méthode que nous avons implémentée dans l'outil *CadesDeviationGenerator* pour faire l'optimisation sous contraintes de fiabilité. Cet outil générique a été appliqué pour optimiser le nano commutateur magnétique avec des contraintes de fiabilité

Conclusion générale et perspectives

Nous avons présenté tout au long de ce rapport les principaux travaux effectués au cours de cette thèse qui se décomposent en trois axes. Chaque axe ouvre de nombreuses perspectives :

Axe modélisation :

Nous avons développé et validé (par une méthode éléments finis) un modèle semi analytique pour calculer la déformation de la poutre avec l'analyse de contact mécanique. Ce modèle a été utilisé pour le couplage magnéto-mécanique dédié pour le nano commutateur magnétique. Ce couplage a été implémenté dans le logiciel MacMMems en prenant en compte des forces d'adhésion. Il a été livré aux autres partenaires du projet MONACO.

Perspectives :

- Utilisation du modèle mécanique pour d'autre dispositif comme le nano résonateur magnétique [DUR-08].
- Comparer le couplage magnéto-mécanique avec la méthode des éléments finis [GAL-11].

Axe dérivation automatique de code :

Nous avons développé un outil de dérivation automatique de code en Java (JAP- Java Jacobian Automatic Programming). JAP surmonte toutes les limitations des dérivateurs actuels en Java que nous avons rencontrés dans notre étude bibliographique, comme la prise en compte des tableaux ou des fonctions externes, ou encore la possibilité de dériver sélectivement. Les performances et la robustesse de JAP ont été comparés avec l'outil existant ADOL-C. JAP a été utilisé dans cette thèse et par d'autres utilisateurs pour dériver des modèles de dimensionnement dans CADES, pour calculer le Jacobien dans l'outil de simulation dynamique développé au G2Elab. Grâce au JAP, nous avons enrichi le framework CADES avec deux générateurs supplémentaires : *JAP_ExternalFunctionGenerator* pour des fonctions externes sous forme algorithmes, *JAPGénérateur* pour des modèles de dimensionnement décrits par des équations sous la forme de langage *sml*.

Perspectives :

- Dérivées d'ordre supérieur (Hessien par exemple) par le développement des class *Jdouble2*, etc.

- Dériver en mode inverse : la version actuelle de JAP dérive en mode direct pour sa facilité d'implémentation. Il a également l'avantage d'être plus économique en mémoire et peu s'avérer plus rapide si le nombre des sorties est supérieur au nombre des entrées. Par contre, si le nombre de sorties est inférieur au nombre des entrées le mode inverse est plus avantageux.
- Appliquer JAP pour FGOT et pour des autres outils métiers du framework CADES comme *Reluctool* développé dans la thèse de B. du Pelloux [DUP-06] et de P.T. Do [DO-10].

Axe optimisation :

Nous avons implémenté des outils d'analyse de sensibilité et d'analyse d'incertitude dans le *CadesCalculator* : dérivé du premier ordre, différence finie, Monte-Carlo. Cet outil a été utilisé pour analyser le nano commutateur magnétique. Grâce à cette étude, nous avons proposé une configuration moins sensible de ce dispositif pour passer à l'étape de l'optimisation. Nous avons aussi appliqué différentes stratégies pour l'optimisation sous contraintes du nano commutateur magnétique.

Une nouvelle approche pour tenir compte de l'incertitude dans l'optimisation a été aussi développée et implémenté dans l'outil *CadesDeviationGenerator* du framework CADES. Cet outil générique sert à faire l'optimisation sous contraintes de fiabilité et a été appliqué pour le nano commutateur magnétique.

Perspectives :

- La méthode de fiabilité que nous avons utilisée est basée sur l'approche N-sigma (6-sigma par exemple). On pourrait appliquer des méthodes plus sophistiquées (FORM, SORM [SAL-00]) pour propager les incertitudes et faire l'optimisation sous contraintes de fiabilité.

Références bibliographiques

- [AKO-84] G. Akoun, J. P. Yonnet. « *3D analytical calculation of the forces exerted between two cuboidal magnets* ». IEEE Transactions on Magnetics Vol. Mag-20 No. 5, pp. 1962-1964, September 1984.
- [ALB-04] L. Albert. « *Modélisation et optimisation des alternateurs à griffes. Application au domaine automobile* ». Mémoire de Thèse - Laboratoire d'Electrotechnique de Grenoble, Institut National Polytechnique de Grenoble, France, 2004.
- [ALL-03] L. Allain. « *Capitalisation et traitement des modèles pour la conception en génie électrique* ». Mémoire de Thèse - Laboratoire d'Electrotechnique de Grenoble, Institut National Polytechnique de Grenoble, France, 2003.
- [ALO-97] P. Alotto, M. Gaggero, G. Molinari and M. Nervi. « *A design of experiment and statistical approach to enhance the generalised response surface method in the optimization of multim minima problems* ». IEEE Transactions on Magnetics, vol. 33, no. 2, pp. 1896-1899, 1997.
- [ANS] Ansys ®. <http://www.ansys.com/>
- [ANS-05] Ansys. « *ANSYS Couples-Field analysis guide* ». 2005.
- [ATI-03] E. Atienza. « *Méthodologie et outil pour le dimensionnement* ». Mémoire de thèse – Laboratoire d'Electrotechnique de Grenoble, Institut National Polytechnique de Grenoble, France, 2003.
- [AUT] Community Portal for Automatic Differentiation.
<http://www.autodiff.org/>
- [BAL-03] R. Balian. « *Effet Casimir et géométrie* ». Séminaire Poincaré (Paris, 9 mars 2002), publié dans : Bertrand Duplantier et Vincent Rivasseau (Eds.); Poincaré Seminar 2002, Progress in Mathematical Physics 30, Birkhäuser (2003).
- [BAO-10] Y. Bao et al. « *The Casimir force on a surface with shallow nanoscale corrugations : Geometry and finite conductivity effects* ». Phys.Rev.Lett. 105(25), 250402 (2010).
- [BAR-08] T. Baron. « *Développement sur substrat SOI mince de composants N-MEMS de type capteur inertiel et étude de la co-intégration avec une filière CMOS industrielle* ». Mémoire de thèse de l'Université Joseph Fourier, Grenoble 1, France, avril 2008.
- [BAR-10] B. Barney. « *Introduction to Parallel Computing - Shared Memory* ». Lawrence

- Livermore National Laborator.
https://computing.llnl.gov/tutorials/parallel_comp/
- [BEN-96] C. Bendtsen and O. Stauning. « *FADBAD, a flexible C++ package for automatic differentiation* ». Technical Report IMM-REP-1996-17, Technical University of Denmark, IMM, Departement of Mathematical Modeling, Lyngby, 1996.
- [BIL-08] L. Bilhaut, P. Andreucci, L. Duraffourg et B. Viala. « *Nano-commutateur magnétique bistable* ». Brevet déposé sous le numéro d'enregistrement 08 52061 par le CEA le 28 mars 2008.
- [BIL-09] L. Bilhaut. « *Actionnement magnétique à l'échelle nanométrique* ». Mémoire de thèse de l'Université Joseph Fourier, Grenoble 1, France, novembre 2009.
- [BIS-96] C. Bischof, A. Carle, P. Khademi, and A. Mauer. « *Adifor 2.0 : Automatic differentiation of fortran 77 programs* ». IEEE Computational Science and Engineering, Volume 3, No.3, 1996.
- [BIS-97] C. Bischof, L. Roh, and A. Mauer. « *ADIC - An extensible automatic differentiation tool for ANSI-C* ». Software-Practice and Experience, 27(12) : 1427-1456, 1997.
- [BUC-04] H. M. Bücker, A. Rasch, and A. Wolf. « *A Class of OpenMP Applications Involving Nested Parallelism* ». In Proceedings of the 19th ACM Symposium on Applied Computing, Nicosia, Cyprus, March 14–17, 2004, volume 1, pages 220–224, New York, 2004. ACM Press.
- [CAL-01] M. Caldora Costa. « *Optimisation de dispositifs électromagnétiques dans un contexte d'analyse par la méthode des éléments finis* ». Mémoire de Thèse - Laboratoire d'Electrotechnique de Grenoble, Institut National Polytechnique de Grenoble, France, 2001.
- [CAP-02] O. Caprani, K. Madsen et H.B. Nielsen. « *Introduction to interval analysis* ». TU Dresden, 2002. <http://www2.imm.dtu.dk/~km/Docs/imm1462.pdf>
- [CAS-48] H. B. G. Casimir and D. Polder. « *The Influence of Retardation on the London-van der Waals Forces* ». Phys. Rev. 73, 360–372 (1948).
- [CHE-07] H. L. Chetouani. « *Microsystèmes et micromanipulation à lévitation diamagnétique conception, réalisation et application a la microfluidique digitale et à la biologie* ». Mémoire de Thèse - Laboratoire d'Electrotechnique - Grenoble, Institut National Polytechnique de Grenoble, France, 2007.
- [CHI-01] C. Chillet and J. Y. Voyant. « *Design-oriented analytical study of a linear electromagnetic actuator by means of a reluctance network* ». IEEE Transactions On Magnetics, Volume 37, No. 4, July 2001.

- [CLE-99] A.N. Cleland et M.L. Roukes « *Nanoscale Mechanics* ». Proceedings of the 24th International Conference on the Physics of Semiconductors (World Scientific Press, Singapore) 392, 261, 1999.
- [COL-00] T. F. Coleman et A. Verma. « *ADMIT-1 : Automatic differentiation and MATLAB interface toolbox* ». ACM Trans. Math. Softw, 26(1) : 150–175, 2000.
- [COL-10] M. Collonge. « *Étude, réalisation et caractérisation de transistors MOS à grille suspendue* ». Mémoire de Thèse - Institut National Polytechnique de Grenoble, France, mars 2010
- [COM] Comsol®. <http://www.comsol.com/>
- [COR] G. F. Corliss and A. Griewank. « *Operator overloading as an enabling technology for automatic differentiation* ». Technical report, Center for Research on Parallel Computation, Rice University, Huston, U.S.
- [COU] J. L. Coulomb. Featuring a Genuine Optimization Tool -FGOT®.
<http://forge-mage.g2elab.grenoble-inp.fr/project/got>
- [COU-02] J. L. Coulomb. « *Optimisation* », chapitre 8 de « *Electromagnétisme et problèmes couplés* », « *Electromagnétisme et éléments finis 3D* ». EGEM, Hermes, 2002.
- [COV] CoventorWare®. <http://www.coventor.com/>
- [DAV-07] S. Da-Veiga. « *Analyse d'incertitudes et de sensibilité : Application aux modèles de cinétique chimique* ». Mémoire de Thèse – de l'université Toulouse III, France, 2007.
- [DEM-09] L. H. De Medeiros, G. Reyne, and G. Meunier. « *Comparison of global force calculations on permanent magnets* ». IEEE Transactions On Magnetics, Volume 34, No. 5, 1998.
- [DEL-03] B. Delinchant. « *Un Environnement à base des Composants, Intégrant le Concepteur et ses Outils, pour des Nouvelles Méthodes de CAO* ». Mémoire de Thèse – Laboratoire d'Electrotechnique - Grenoble, Institut National Polytechnique de Grenoble, France, 2003.
- [DEL-04] B. Delinchant, F. Wurtz, and E. Atienza. « *Reducing sensitivity analysis time-cost of compound model* ». IEEE, Transactions on Magnetics, Volume 40, No. 2, March 2004.
- [DEL-07] B. Delinchant, D. Duret, L. Estrabaut, L. Gerbaud, N. H. Hieu, B. Du Peloux, H. L. Rakotoarison, F. Verdier, and F. Wurtz. « *An optimizer using the software component paradigm for the optimization of engineering systems* ». COMPEL, Volume 26, No. 2, 2007.

- [DEL-09] B. Delinchant, G. Gruosso, and F. Wurtz. « *Two levels modeling for the optimization of electromagnetic actuators* ». IEEE Transactions On Magnetics, Volume 45, No. 3, 2009.
- [DEZ-04] E. Dezille. « *Utilisation de la différentiation de code dans le cadre du dimensionnement sous contraintes* ». Mémoire de master 2 recherche, Institut National Polytechnique, Grenoble, France, 2004.
- [DO-10] P.T. Do. « *Simulation dynamique des actionneurs et capteurs électromagnétiques par réseaux de réductances : modèles, méthodes et outils* ». Mémoire de Thèse - Grenoble Génie Electrique Laboratoire, Institut National Polytechnique de Grenoble, France, 2010.
- [DUB-11] V. Dubourg, B. Sudret, J.M. Bourinet et M. Cazuguel. « *Optimisation sous contrainte de fiabilité d'une coque imparfaite* ». 10^e Colloque national en calcul des structures, 9-13 mai 2011, Presqu'île de Giens (Var).
- [DUP-03] B. Duplantier. « *Introduction à l'effet Casimir* ». Séminaire Poincaré (Paris, 9 mars 2002), publié dans : Bertrand Duplantier et Vincent Rivasseau (Eds.); Poincaré Seminar 2002, Progress in Mathematical Physics 30, Birkhäuser (2003).
- [DUP-06] B. Du Peloux. « *Modélisation des actionneurs électromagnétiques par réseaux des réductances. Création d'un outil métier dédié au prédimensionnement par optimisation* ». Mémoire de Thèse - Laboratoire d'Electrotechnique de Grenoble, Université Joseph Fourier, France, 2006.
- [DUR-02] A. Dürrbaum, W. Klier and H. Hahn. « *Comparison of automatic and symbolic differentiation in mathematical modeling and computer simulation of rigid-body systems* ». Multibody System Dynamics, Volume 7 No 4, May 2002.
- [DUR-06] L. Duraffourg, P. Andreucci. « *Casimir force between doped silicon slabs* ». Physics Letters A, Vol. 359, pp. 406-411, 2006.
- [DUR-07] D.Duret, L. Gerbaud, F. Wurtz, J.P. Keradec, B. Cogitore, « *Modeling of passive electronic circuits with sensitivity analysis dedicated to the sizing by optimization* ». KES'07 Proceedings of the 11th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, p422-430.
- [DUR-08] L. Duraffourg, P. Andreucci, L. Bilhaut et B. Viala. « *Nano résonateur magnétique* ». brevet déposé sous le numéro d'enregistrement 08 52062 par le CEA le 28 mars 2008.
- [DYM] Dymola®, <http://www.3ds.com/products/catia/portfolio/dymola>

- [ELS-08] A. Elsheim, S. Noack and W. Wiechert. « *Sensitivity analysis of modelica applications via automatic differentiation* ». In 6th International Modelica Conference, Volume : 2, On page(s) : P669-P675, 2008.
- [ENC-09-1] P. Enciu, F. Wurtz, L. Gerbaud, and B. Delinchant. « *Automatic differentiation for electromagnetic models used in optimization* ». COMPEL : The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, Volume 28, No.5, 2009.
- [ENC-09-2] P. Enciu. « *Dérivation automatique pour le calcul des sensibilités appliqué au dimensionnement en génie électrique* ». Mémoire de Thèse - Grenoble Génie Electrique Laboratoire, Institut National Polytechnique de Grenoble, France, 2009.
- [FOR-06] S. A. Forth. « *An efficient overloaded implementation of forward mode automatic differentiation in MATLAB* ». ACM Transactions on Mathematical Software, 32(2) : 195–222, June 2006.
- [FMI] Fmincon, Matlab.
<http://www.mathworks.fr/help/toolbox/optim/ug/fmincon.html>
- [FIS-04] V. Fischer. « *Composants Logiciels pour le Dimensionnement en Génie Electrique. Application à la Résolution d'Equations Différentielles* ». Mémoire de Thèse - Laboratoire d'Electrotechnique de Grenoble, Institut National Polytechnique de Grenoble, France, 2004.
- [FIS-05] V. Fischer, L. Gerbaud and F. Wurtz. « *Using automatic code differentiation for optimization* ». IEEE, Transactions on Magnetics, Volume 41, No. 5, May 2005.
- [GAL-11] N. Galopin, P. Pham-Quang, B. Delinchant and J.L. Coulomb. « *Finite Element Modeling of Deformable NEMS taking into account mechanical contact* ». Compumag 2011, Sydney, Australia, 11-15 July 2011.
- [GIA-08] Y. B. Gianchandani, O. Tabata, H. Zappe. « *Comprehensive Microsystems* ». Elsevier, 2008.
- [GIE-05] R. Giering, T. Kaminski et T. Slawig. « *Generating Efficient Derivative Code with TAF : Adjoint and Tangent Linear Euler Flow Around an Airfoil* ». Future Generation Computer Systems, 21(8) : 1345–1355, 2005
- [GRI-08] A. Griewank and A. Walther. « *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation* ». SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2008.
- [HAL-00] A. Haldar and S. Mahadevan. « *Probability, Reliability, and Statistical*

- Methods in Engineering Design* ». Wiley, New York, 2000.
- [HAS-04] L. Hascoët and V. Pascual. « *Tapenade 2.1 user's guide* ». Rapport Technique, Institut National de la Recherche en Informatique et en Automatique - INRIA, France, 2004.
- [HAS-05] L. Hascoët. « *Analyses statiques et transformations de programmes : de la parallélisation à la différentiation* ». Mémoire de habilitation à diriger des recherches, Université de Nice Sophia-Antipolis, France, 2005.
- [INT] IntelliSuite®. <http://intellisensesoftware.com/>
- [JAC-05] J. Jacques. « *Contributions à l'analyse de sensibilité et à l'analyse discriminante généralisée* ». Mémoire de Thèse – de l'Université Joseph Fourier - Grenoble 1, France, 2005.
- [JAN-09] J. L. G. Janssen, J. J. H. Paulides, and E. A. Lomonova, « *Passive magnetic suspension limitations for gravity compensation* ». Journal of System Design and Dynamics, vol. 3, no. 4, pp. 671–680, 2009.
- [JAN-11] J.L.G. Janssen, J.J.H. Paulides, E.A. Lomonova, B. Delinchant, J.-P. Yonnet, « *Design study on magnetic springs with low resonance frequency* » International Symposium on Linear Drives for Industry Applications (LDIA 2011) July 3-6 2011 LDIA 2011, Eindhoven, Netherlands.
- [JYT] Jython. <http://www.jython.org/>
- [KEN-95] J. Kennedy and R. Eberhart. « *Particle swarm optimization* ». In Proceedings of IEEE International Conference on Neural Networks, volume IV, pages 1942-1948, Perth, Australia, 1995.
- [KOF] M. Kofler. « *Maple : An introduction and references* ». Addison Wesley
- [KOW-07] A. Kowarz. « *Advanced concepts for automatic differentiation based on operator overloading* ». Ph.D. thesis, TU Dresden, 2007.
- [LAM-00] A. Lambrecht and S. Reynaud, Eur. Phys. J. D8, 309 (2000)
- [LAM-03] A. Lambrecht et S. Reynaud « *Recent experiments on the Casimir effect : description and analysis* ». Séminaire Poincaré (Paris, 9 mars 2002), publié dans : Bertrand Duplantier et Vincent Rivasseau (Eds.); Poincaré Seminar 2002, Progress in Mathematical Physics 30, Birkhäuser (2003).
- [LAM-08] A. Lambrecht and V. N. Marachevsky. « *Casimir Interaction of Dielectric Gratings* ». Phys.Rev.Lett, 101, 160403 (2008).
- [LAM-10] A. Lambrecht A. Canaguier-Durand, R. Guérout et S. Reynaud. « *Casimir effect in the scattering approach : correlations between material properties, temperature and geometry* ». arXiv :1006.2959 (2010).

- [MAC] MacMMems®.
<http://forge-mage.g2elab.grenoble-inp.fr/project/macmmems>
- [MAG-04] D. Magot. « *Méthodes et Outils Logiciels d'aide au Dimensionnement. Application aux composants Magnétiques et aux filtres passifs* ». Mémoire de Thèse – Laboratoire d'Electrotechnique de Grenoble, Institut National Polytechnique de Grenoble, France, 2004.
- [MOD] Modelica. <http://www.modelica.org/>
JModelica. <http://www.jmodelica.org/>
- [MOK-11] L. Mokhtari, B. Delinchant, J-L. Coulomb, T. Le duc, and P. Pham Quang. « *Software Coupling and Orchestration Tool to the Modeling of Multi-physic Problems* ». IEEE Transactions On Magnetics (submitted).
- [MEM] MEMS Pro®. <http://www.softmems.com/>
- [NAT-67] H. C. Nathanson, W. E. Newell, R. A. Wickstrom, J. R. Jr. Davis. « *The resonant gate transistor* », IEEE Transactions on Electron Devices, Vol. 14, No. 3, pp. 117-133, 1967.
- [NIA-03] D. Niarchos. « *Magnetic MEMS : key issues and some applications* », Sensors and Actuators A, vol. 109, Issues 1-2, 1 December 2003, Pages 166-173.
- [NGU] H. Nguyen Huu. Travaux de posdoct dans le cadre du projet MEGEVE en partenariat avec Thales (confidentiel).
- [NGU-10] H. Nguyen Xuan, J.L. Coulomb, L. Gerbaud, J.C. Crebier. « *Application of progressive quadratic response surface method for an oscillation problem optimization* ». OIPE 2010, 14-18 Septembre 2010, Sofia, Bulgaria.
- [OCT] Octave®. <http://www.gnu.org/software/octave/>
- [OOF] Oofelie®. <http://www.open-engineering.com/>
- [PAR-05] K. Park et S. Moon. « *Optimal design of heat exchangers using the progressive quadratic response surface model* ». International Journal of Heat and Mass Transfer, vol. 48, no. 11, pp. 2126-2139, May 2005.
- [PEN-09] F. Pennec. « *Modélisation du contact metal-metal : application aux microcommutateur MEMS RF* ». Mémoire de Thèse - Université Paul Sabatier- Toulouse III, France, juin, 2009.
- [PET] R. J. Petti. « *Introduction to macsymba* ». Publishers, Inc.
- [PET-82] K. E. Petersen. « *Silicon as a mechanical material* ». Proceedings of the IEEE, Vol. 70, No. 5, pp. 420-457, 1982.
- [PEY-03] J.F. Peyrucat. « *Le Six Sigma, une démarche pour traquer les défauts du process* ». Dans Mesures, no 753, mars 2003, p. 20.

- [PHA] P. Pham-Quang, H. L. Rakotoarison et B. Delinchant « *MacMMems- Manuel d'utilisation* ». (document téléchargeable)
<http://forge-mage.g2elab.grenoble-inp.fr/project/macmmems/browse/trunk/lib/MacMMemsManuelUtilisation.doc>
- [PHA-10-1] P. Pham Quang, B. Delinchant, J.L. Coulomb and B. du Peloux « *Semi-analytical Magnetic-Structural coupling with contact analysis for MEMS/NEMS* ». CEFC 2010, Chicago, IL USA – May 9th-May 12th 2010.
- [PHA-10-2] P. Pham Quang, B. Delinchant, J.L. Coulomb and B. du Peloux. « *Optimization methodology of a magnetic deformable nano switch based on semi-analytical model* ». OIPE 2010, 14-18 Septembre 2010, Sofia, Bulgaria.
- [PHA-11-1] P. Pham Quang, B. Delinchant, J.L. Coulomb and B. du Peloux. « *Semi-Analytical Magneto-Mechanic Coupling With Contact Analysis for MEMS/NEMS* ». IEEE Transactions On Magnetics, Volume 47, No. 5, May 20011.
- [PHA-11-2] P. Pham-Quang, B. Delinchant, C. Ilie, E. Slusanschi, J.L. Coulomb, and B. du Peloux « *Mixing Techniques to Compute Derivatives of semi-numerical models : Application to Magnetic Nano Switch Optimization* ». Compumag 2011, Sydney, Australia, 11-15 July 2011.
- [PRE-91] W. H. Press and S. A. Teukolsky. « *Numerical calculation of derivatives* ». Computers in Physics, 5(1) : 88-89, Jan. /Feb 1991.
- [PRE-04] I. L. Prejbeanu, W. Kula, K. Ounadjela, R.C. Sousa, O. Redon, B. Dieny et J.P. « *Nozières Thermally Assisted Switching in Exchange Biased Storage Layer Magnetic Tunnel Junctions* ». IEEE Transactions on Magnetics, Vol. 40, No. 4, pp. 2625-2627, 2004.
- [PRE-07] I. L. Prejbeanu, M. Kerekes, R. C. Sousa, H. Sibuet, O. Redon, B. Dieny et J. P. Nozières. « *Thermally assisted MRAM* ». Journal of Physics Condensed Matter, Vol. 19, 165218, 2007.
- [RAC-01] R. Rackwitz. « *Reliability analysis – a review and some perspectives* ». Structural Safety 23 (2001), pp. 365–395.
- [RAD] Radia®.
<http://www.esrf.eu/Accelerators/Groups/InsertionDevices/Software/Radia>
- [RAK-07] H. L. Rakotoarison. « *Méthode et Outil de Génération Automatique de modèle pour l'optimisation Fortement Contrainte des Microsystèmes Magnétiques* ». Mémoire de Thèse - Laboratoire d'Electrotechnique de Grenoble,

- Université Joseph Fourier, France, 2007.
- [REY-02] G. Rey, J. Delamare, O. Cugat. « *Microactionneur électromagnétiques-MAGMAS* ». Collection Hermes- Science / Lavoisier ISBN 2- 7462-0449-5, 2002.
- [ROS-83] J.H. Rose, J.R. Smith, J. Ferrante, Phys. Rev. B28, 1835 (1983)
- [SAL-00] A. Saltelli, K. Chan and E. M. Scott. « *Sensitivity Analysis* ». Chichester : Wiley Series in Probability and Statistics, 2000.
- [SAL-04] A. Saltelli, S. Tarantola, F. Campolongo and M. Ratto. « *Sensitivity Analysis in Practice* ». Chichester : Wiley, 2004.
- [SAL-08] A. Saltelli, M. Ratto, T. Andres and F. Campolongo. « *Global Sensitivity Analysis* ». The Primer : Wiley, 2008.
- [SCI] Scilab®. <http://www.scilab.org/>
- [SEN-11] B. Sensanga. « *Optimisation de Gammes : Application à la Conception des Machines Synchrones à Concentration de Flux* ». Mémoire de Thèse - Grenoble Génie Electrique Laboratoire, Institut National Polytechnique de Grenoble, France, 2011.
- [SHI-96] D. Shiriaev. « *ADOL-F automatic differentiation of Fortran codes* ». In M. Berz, C. H. Bischof, G. F. Corliss, and A. Griewank, editors, *Computational Differentiation : Techniques, Applications, and Tools*, pages 375–384. SIAM, Philadelphia, PA, 1996.
- [SKJ-01] R. Skjelvic. « *Automatic differentiation in java* ». Master's thesis, Department of Informatics, University of Bergen, Norway, 2001.
- [SLU-08] E.I. Slusanschi. « *Algorithmic differentiation of Java programs* ». Ph.D. thesis, RWTH Aachen University, 2008.
- [TAP] TAPENADE en ligne. <http://tapenade.inria.fr:8080/tapenade/>
- [TIM-49] S. P. Timoshenko. « *Résistance des matériaux* ». Ed. Dunod, 1949 (1^{ère} édition en anglais).
- [TSO-08] Y. Tsompanakis, N.D. Lagaros, and M. Papadrakis, editors. « *Structural design optimization considering uncertainties* ». Taylor & Francis, 2008.
- [VIA-08] B. Viala. « *Le ferromagnétisme en couche mince pour les hyperfréquences* ». Manuscrit d'habilitation à diriger les recherches, Université Joseph Fourier, 2008.
- [WAL] A. Walther and A. Griewank. « *Adol-C : A package for the automatic differentiation of algorithms written in C/C++* ». Technical report, Institute of Scientific Computing, Technische Universität Dresden, Dresden, Germany.

- [WAL-07] A. Walther. « *Développement de couches magnétiques dures pour MEMS : application à un microswitch magnétique bistable* ». Mémoire de Thèse - Université Joseph Fourier, France, 2007.
- [WAN-03] Wang et G. Gary. « *Adaptive Response surface method using inherited Latin Hypercube Design points* ». Journal of mechanical design 2003, vol. 125, no2, pp. 210-220.
- [WIK-1] http://fr.wikipedia.org/wiki/Nanosystème_électromécanique
- [WIK-2] http://en.wikipedia.org/wiki/Normal_distribution
- [WIK-3] http://fr.wikipedia.org/wiki/Six_Sigma
- [WOL] S. Wolfram. « *The mathematica book* ». Cambridge University Press, 4th Edition.
- [ZAN-07] A. Zaneti. « *Modélisation et simulation de microsystemes multi domaines à mixtes : vers le prototypage virtuel d'un microsysteme autonome* ». Mémoire de Thèse - Université Joseph Fourier, France, 2007.

Publications

Articles de revues internationales avec comité de relecture

IEEE, Transactions on Magnetics,
May 2011. « *Semi-Analytical Magneto-Mechanic Coupling With Contact Analysis for MEMS/NEMS* »
P. Pham Quang, B. Delinchant, J.L. Coulomb and B. du Peloux.

Abstract : This paper presents a methodology and a tool for magnetic and mechanical deformation coupling using numerical and analytical modeling. An analytical magnetic model using Coulombian approach is used and coupled with a mechanical deformation model for a cantilever beam to evaluate contact size and contact force. Such a coupling is not available using numerical solution. This paper details the deformation and contact analysis, which is validated by finite element simulation and also details the coupling approach. Such a modeling is dedicated to an optimization process of magnetic MEMS/NEMS in general and to magnetic nano switch in particular.

Articles de conférences internationales avec comité de relecture

- CEFC 2010,
Chicago, IL USA. « *Semi-analytical Magnetic-Structural coupling with contact analysis for MEMS/NEMS* »
P. Pham Quang, B. Delinchant, J.L. Coulomb and B. du Peloux.
- OIPE 2010,
Sofia, Bulgaria. « *Optimization methodology of a magnetic deformable nano switch based on semi-analytical model* »
P. Pham Quang, B. Delinchant, J.L. Coulomb and B. du Peloux.
- Compumag 2011,
Sydney, Australia. « *Mixing Techniques to Compute Derivatives of semi-numerical models : Application to Magnetic Nano Switch Optimization* »
P. Pham-Quang, B. Delinchant, C. Ilie, E. Slusanschi, J.L. Coulomb, and B. du Peloux.
- Compumag 2011,
Sydney, Australia. « *Finite Element Modeling of Deformable NEMS taking into account mechanical contact* »
N. Galopin, **P. Pham-Quang**, B. Delinchant and J.L. Coulomb.

Annexes

A. Jap Exceptions

Nom d'exception	Description d'Exception/Contre Exemple
Jap Exception 01	Jap keywords : "jap_", "jacobian_", "Jdouble", "partialDerivative_" has been found
	<pre>public double func1(double jap_x1, double x2){... }</pre>
Jap Exception 02	Unsupported data type : Data type accepted : <i>boolean, double, double[], double[[[]], int, int[], int[[[]]</i> .
	<pre>public double func1(long t1, double[[[]]] x){... }</pre>
Jap Exception 03	Overloading method Overloading method with the same number of parameter is not allowed
	<pre>public double func1(double x1, double x2){... } public double func1(int t1, double x2){... }</pre>
Jap Exception 04	Jap cannot accept declaration "new" (for declaration table) in types of instruction : scalar operation <i>for, if, while, ...</i>
	<pre>b=(new double[2]).length*x1; for(i=0;i<(new double[2]).length; i++)</pre>
Jap Exception 05	Jap can not accept : instruction with ")]", "=)"
	<pre>(b)=x1+1; b=func1(x1, x2)[1] ;</pre>
Jap Exception 06	Variables must be declared immediately after the signature of the method WITHOUT assignment.
	<pre>double b=x1+1 ;</pre>
Jap Exception 07	No transformation data (<i>double</i>), (<i>double[]</i>), (<i>double[[[]]</i>) is allowed
	<pre>b=(double)t1+1 ;</pre>

Jap Exception 08	<p>Problem of index of table and vector :</p> <ul style="list-style-type: none">cannot accept appearance of <i>(new),(int)</i>cannot accept appearance of methodcannot accept appearance of variable <i>double, double[], double[][]</i>cannot accept appearance of variable <i>int[],int[][]</i> for index of table and vector of int
Jap Exception 09	<p>Jap cannot accept appearance of <i>(new, int, double, boolean)</i> in instruction with calling method or return.</p>

B. Services de calcul de sensibilité dans CadesCalculator

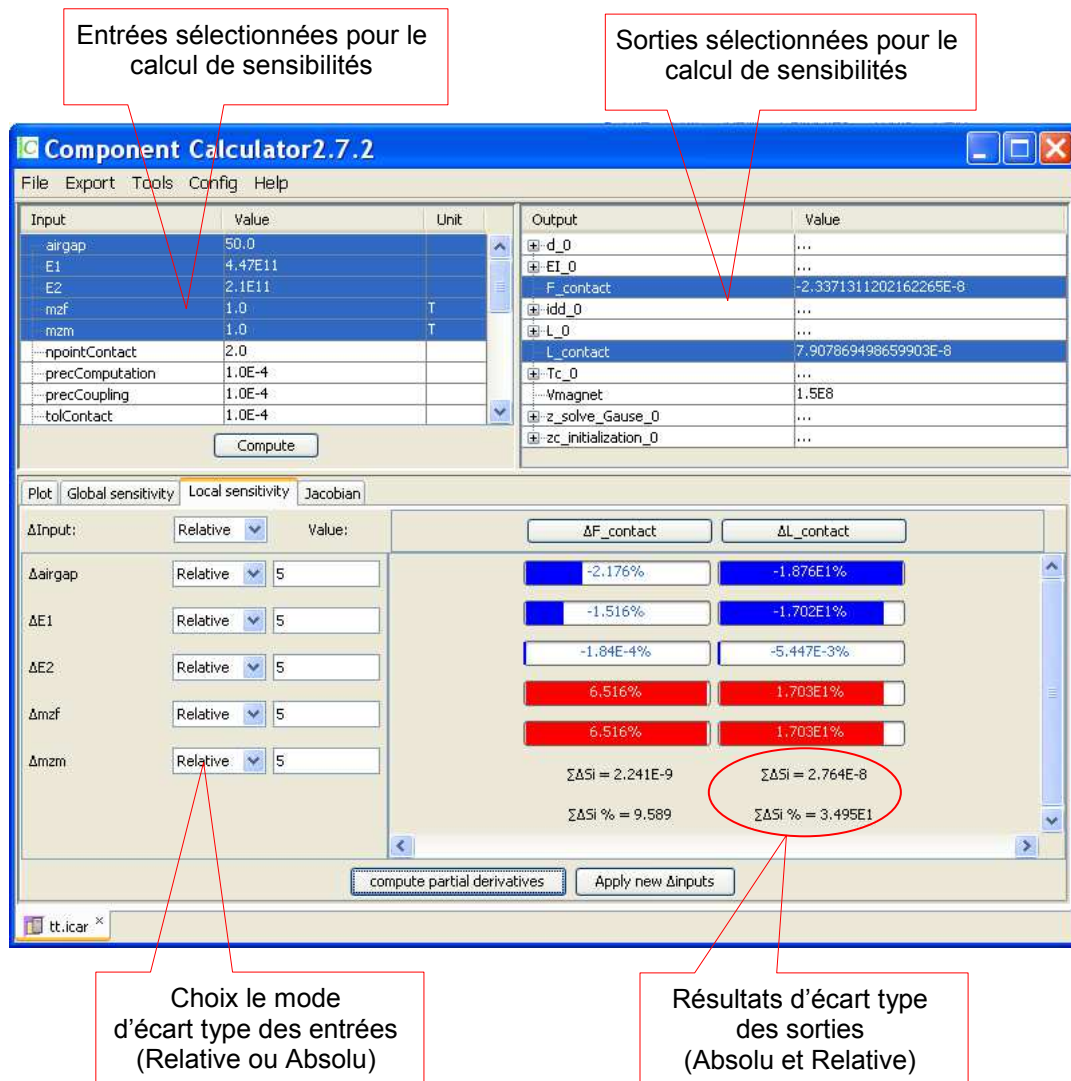


Figure B-1. L'interface graphique du service de calcul de sensibilité locale avec la dérivée de premier ordre

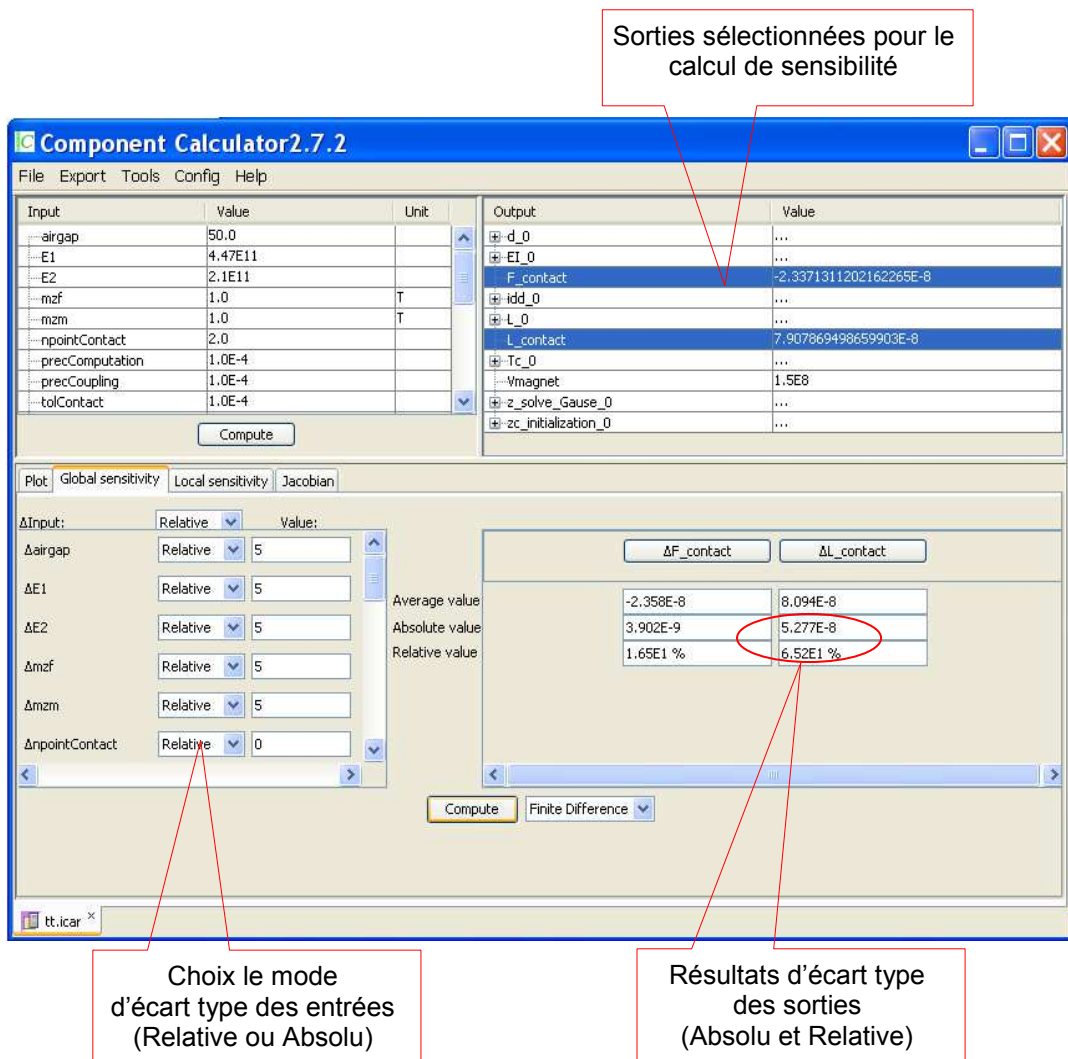


Figure B-2. L'interface graphique du service de calcul de sensibilité avec la méthode différences finies

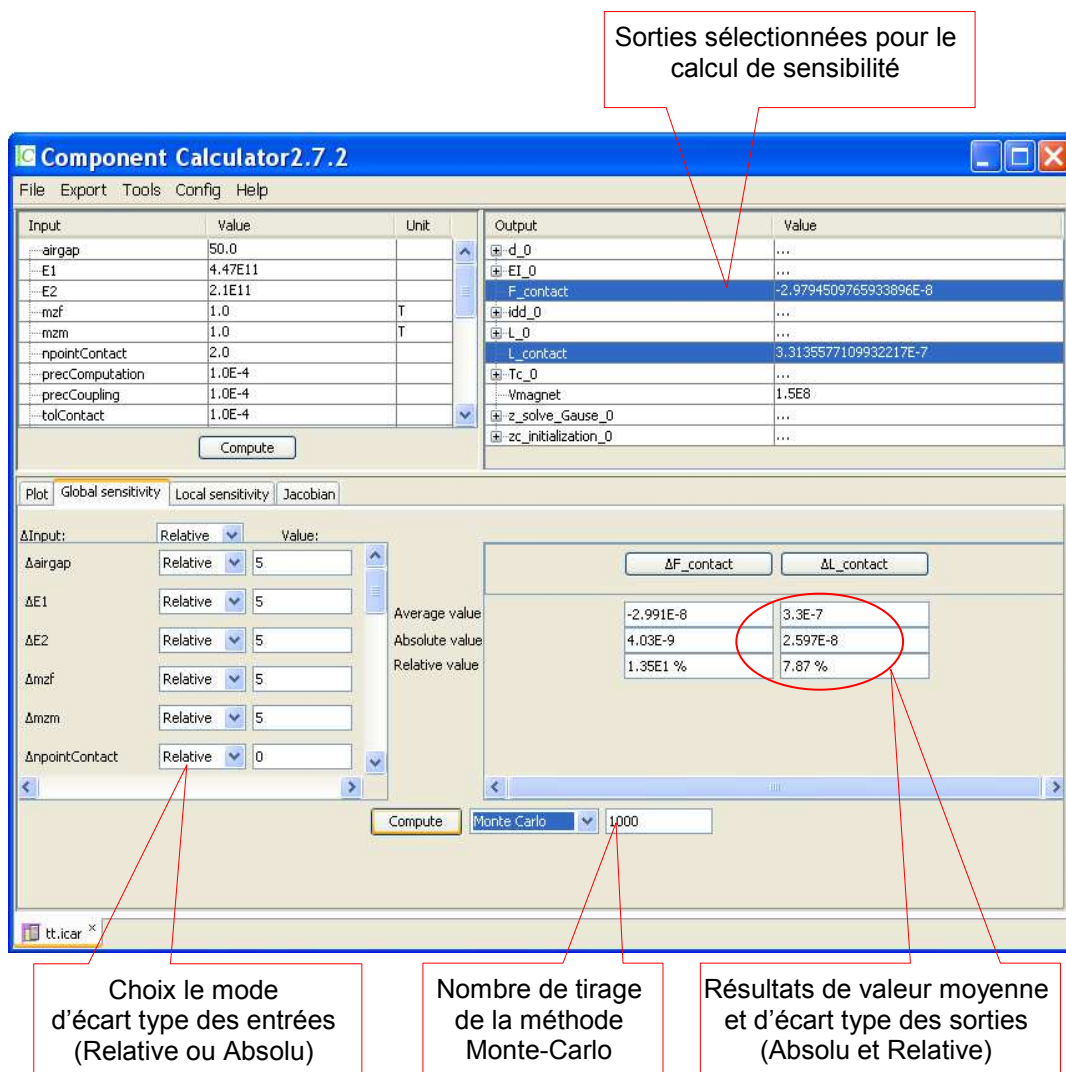


Figure B-3. L'interface graphique du service de calcul de sensibilité globale avec la méthode Monte-Carlo

C. Géométries et paramétrage du nano commutateur magnétique

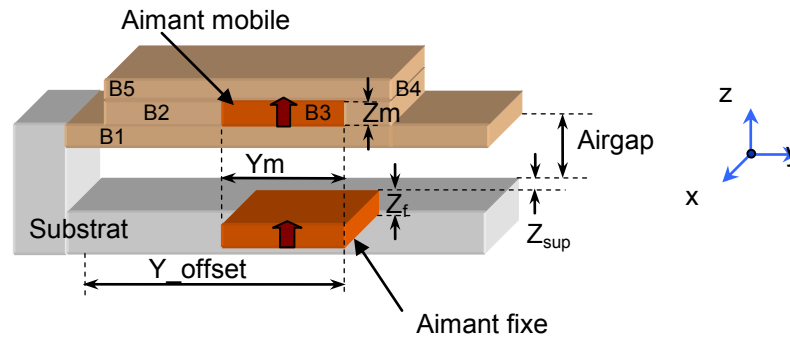


Figure C-1. Géométrie initiale du nano commutateur magnétique

Paramètres	Explications	Valeurs	unité
Mz	Magnétisation verticale des aimants	2,4	T
Airgap	Air gap	50	nm
Z _{sup}	Substrat entre l'aimant fixe et la surface de contact	10	nm
E1	Module d'Young de l'aimant (Ru-FeMn-FeCo)	2,1x10 ¹¹	Pa
E2	Module d'Young de la poutre (Pt)	1,68x10 ¹¹	Pa
X1	Largeur de la poutre (B1÷B5)	500	nm
Z1	Hauteur de B1	20	nm
Zm	Hauteur de B2÷B4	300	nm
Zf	Hauteur de l'aimant fixe	300	nm
Z5	Hauteur de B5	50	nm
Y1	Longueur de B1	4000	nm
Y2	Longueur de B2	2500	nm
Ym	Longueur de B3 (aimant fixe et mobile)	500	nm
Y4	Longueur de B4	500	nm
Y_offset	Position des aimants sur la poutre	3000	nm

Tableau C-1. Paramétrages de géométrie initiale du nano commutateur magnétique

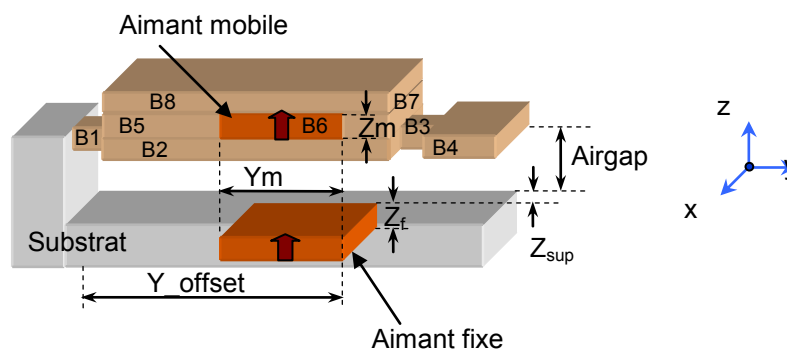


Figure C-2. Géométrie améliorée du nano commutateur magnétique

Paramètres	Explications	Valeurs	unité
Mz	Magnétisation verticale des aimants	2,4	T
My	Magnétisation horizontale des aimants	2,4	T
Airgap	Air gap	50	nm
Z _{sup}	Substrat entre l'aimant fixe et la surface de contact	10	nm
E1	Module d'Young de l'aimant (Ru-FeMn-FeCo)	2,1x10 ¹¹	Pa
E2	Module d'Young de la poutre (Pt)	1,68x10 ¹¹	Pa
X1	Largeur de B1, B3	100	nm
X2	Largeur de la poutre sauf B1, B3	500	nm
Z1	Hauteur de B2÷B4	20	nm
Zm	Hauteur de B5÷B7	300	nm
Zf	Hauteur de l'aimant fixe	300	nm
Z8	Hauteur de B8	50	nm
Y1	Longueur de B1	100	nm
Y2	Longueur de B2, B8	3400	nm
Y3	Longueur de B3	200	nm
Y4	Longueur de B4	300	nm
Ym	Longueur de B6 (aimant fixe et mobile)	500	nm
Y_offset	Position des aimants sur la poutre	3000	nm

Tableau C-2. Paramétrages de géométrie améliorée du nano commutateur magnétique

D. Résultats de comparaison entre JAPGenerator et AdolCGenerator sur les cas test de CADES

	unité	AdolCGenerator	JAPGenerator non sélective	JAPGenerator sélective
Taille de ICAr	[KB]	1041	214	216
Temps de génération	[s]	10,3	12,5	12,5
Nombre d'itération		7	7	7
Temps d'optimisation	[s]	0,4	0,4	0,4

Tableau D-1. Comparaison entre JAPGenerator et AdolCGenerator sur l'application de machines à concentration de flux [SEN-11]

	unité	AdolCGenerator	JAPGenerator non sélective	JAPGenerator sélective
Taille de ICAr	[KB]	755	173	175
Temps de génération	[s]	4,4	3,9	3,9
Nombre d'itération		73	73	73
Temps d'optimisation	[s]	3,5	4,0	4,1

Tableau D-2. Comparaison entre JAPGenerator et AdolCGenerator sur l'application d'actionneur linéaire [CHI-01]

	unité	AdolCGenerator	JAPGenerator non sélective	JAPGenerator sélective
Taille de ICAr	[KB]	979	114	116
Temps de génération	[s]	16,1	18,2	18,2
Nombre d'itération		64	64	64
Temps d'optimisation	[s]	2,9	4,2	3,5

Tableau D-3. Comparaison entre JAPGenerator et AdolCGenerator sur l'application d'alternateur à griffe [ALB-04]

	unité	AdolCGenerator	JAPGenerator non sélective	JAPGenerator sélective
Taille de ICAr	[KB]	1490	107	109
Temps de génération	[s]	15,1	4,8	4,8
Nombre d'itération		98	98	98
Temps d'optimisation	[s]	10,5	5,7	2,1

Tableau D-4. Comparaison entre JAPGenerator et AdolCGenerator sur l'application du projet MEGEVE [NGU]

	unité	AdolCGenerator	JAPGenerator non sélective	JAPGenerator sélective
Taille de ICAr	[KB]	769	223	225
Temps de génération	[s]	3,2	2,2	2,2
Nombre d'itération		13	13	13
Temps d'optimisation	[s]	313,9	65,5	24,4

Tableau D-55. Comparaison entre JAPGenerator et AdolCGenerator sur l'application de nano commutateur magnétique (chapitre II)

Modélisation magnéto-mécanique d'un nano commutateur.

Optimisation sous contraintes de fiabilité par dérivation automatique des programmes en Java.

Résumé :

Les nano commutateurs magnétiques sont en cours d'étude et ils sont susceptibles d'être présents dans plusieurs domaines d'application comme les interrupteurs d'alimentation, les convertisseurs DC/DC...etc. Compte tenu de leur très petite taille et de leur procédé de fabrication, leurs performances sont sujettes à de grandes dispersions. Partant du besoin de modélisation et d'optimisation avec fiabilité de ces dispositifs, ces travaux de thèse se décomposent en trois axes.

Axe modélisation : Développement d'un modèle semi analytique pour calculer la déformation avec l'analyse de contact mécanique. Ce modèle a été introduit pour le couplage magnéto-mécanique dans le logiciel de modélisation de MEMS magnétiques MacMMems.

Axe dérivation automatique de code : Développement de JAP (Java Jacobian Automatic Programming) qui est un outil générique de dérivation d'algorithmes. Il a été appliqué pour le nano commutateur et a été introduit dans les outils de dimensionnement développés au G2Elab pour l'analyse de sensibilité, pour l'optimisation exploitant le Jacobien et aussi pour orienter des systèmes d'équations algèbro-différentiels.

Axe optimisation : Mise en œuvre du modèle et des outils développés pour faire l'étude de sensibilité et l'optimisation sous contraintes de fiabilité du nano commutateur magnétique.

Mots Clés : analyse de contact, couplage magnéto-mécanique, nano commutateur magnétique, dérivation automatique de code, analyse de sensibilité, optimisation, fiabilité.

Abstract :

Magnetic nano switches are being studied and they are envisaged in several application areas such as power switches, DC / DC converters ...etc. Given their very small size and their manufacturing process, their performance is subject to large dispersions. Hence the need for modelling and optimization with reliability of these devices, this thesis work is divided into three areas.

Modelling : Development of a semi analytical model to calculate the deformation with the analysis of mechanical contact. This model was introduced in the MacMMems software dedicated to the modelling of magnetic MEMS.

Automatic differentiation: Development of JAP (Java Jacobian Automatic Programming) is a generic algorithms derivation program. It has been applied to the nano switch and was introduced in G2Elab design tools for sensitivity analysis, for optimizing and also to solve differential-algebraic systems.

Optimization: Development of the model and tools to study the sensitivity and reliability-based design optimization for magnetic nano switch.

Keywords : contact analysis, magneto-mechanical coupling, nano magnetic switch, automatic differentiation, sensitivity analysis, optimization, reliability .