EDSP
École Doctorale
Sciences Pratiques

ENS CACHAN

ENSC–2013/482

THÈSE DE DOCTORAT
DE
L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Présentée par

Luc LAURENT

pour obtenir le grade de

DOCTEUR DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Domaine

MÉCANIQUE – GÉNIE MÉCANIQUE – GÉNIE CIVIL

Sujet de la thèse

Stratégie multiparamétrique et métamodèles pour l'optimisation multiniveaux de structures

Soutenue à Cachan le 2 décembre 2013 devant le jury composé de :

Pierre DUYSINX	Université de Liège	& Président du jury Rapporteur
Rodolphe LE RICHE	École des Mines de Saint-Étienne	Rapporteur
Piotr BREITKOPF	Université de Technologique de Compiègne	Examinateur
Stéphane GUINARD	EADS IW, Blagnac	Examinateur
Pierre-Alain BOUCARD	École Normale Supérieure de Cachan	Directeur de Thèse
Bruno SOULIER	École Normale Supérieure de Cachan	Co-encadrant

LMT-Cachan
ENS Cachan / CNRS / UPMC / PRES UniverSud Paris
61, avenue du Président Wilson, F-94235 Cachan cedex, France

Ce manuscrit marque la fin de sept belles années à l'ENS. Que de gens à remercier !

Avant toute chose, un grand merci à Pierre-Alain et Bruno de m'avoir proposé ce sujet. Je les remercie également de leur disponibilité, de leurs conseils avisés et des nombreuses discussions fructueuses durant ces 3 ans et demi de travaux.

Je souhaite ensuite remercier M. Pierre DUYSINX et M. Rodolphe LE RICHE d'avoir accepté de rapporter mon travail. Merci également à M. DUYSINX d'avoir présidé mon jury ainsi qu'à M. Piotr BREITKOPF et M. Stéphane GUINARD d'avoir fait partie de celui-ci en tant qu'examinateurs.

Mille mercis à l'ensemble de l'équipe du centre de calcul, Frisou, Philippe, Pierre, Arnaud, Samir, sans qui tous mes développements informatiques et mes calculs sur le cluster n'auraient pu être menés à terme.

Un grand merci aussi à Lydia pour sa patience, sa gentillesse, sa disponibilité et sa gestion sans faille des déplacements, conférences...

Plus globalement, je remercie l'ensemble du LMT-Cachan pour 3 superbes années en leur compagnie dans un cadre de bonne humeur sans faille.

Sur un autre volet, je remercie le Département de Génie Mécanique de m'avoir confié une mission d'enseignement. Merci en particulier à Danielle, Christine et Christine pour leurs attentions et leur gentillesse. Je souhaite également remercier Laurent, Bruce, Pewee, Étienne et plus globalement l'ensemble de l'équipe enseignante qui, après m'avoir eu comme élève, m'a accueilli comme enseignant.

Je remercie enfin tous ceux qui ont rendu ces 3 années inoubliables : Jérémy, Adrien, Erwan, Pierre, Cécile, Benoît, Anita, Stall, Floflo, Andrea, Geoffrey, Sylvain, Guillaume, Nicolas, Maxime, Christophe, Carole, Enrico, Matteo... Des bons moments en 211, au bar du LMT, de nombreuses soirées... J'en oublie certainement, mais la liste est longue. Merci à tous ! !

Mon plus grand merci va, sans conteste, à Marie pour son soutien sans faille, sa patience (il en a fallu ! !) et son aide pour rendre ce manuscrit ortho-grammatico-conforme.

Table des matières

Table des matières i

Table des figures vii

Liste des tableaux xi

Introduction 1

1 Outils au service de l'optimisation d'assemblages 9
1 Introduction . . . 10
2 Optimisation . . . 12
2.1 Différents types d'optimisation et de stratégies . . . 12
2.1.1 Optimisation de forme . . . 12
2.1.2 Optimisation paramétrique . . . 13
2.2 Formulation d'un problème d'optimisation paramétrique . . . 14
2.2.1 Formulation générale et vocabulaire . . . 14
2.2.2 Solutions du problème d'optimisation et vocabulaire . . . 15
2.2.3 Quelques éléments relatifs à l'unicité et à l'existence du minimum . . . 17
2.2.4 Problématiques de l'optimisation . . . 17
2.3 Algorithmes d'optimisation . . . 19
2.3.1 Optimisation locale ou globale ? . . . 19
2.3.2 Algorithmes d'optimisation locale . . . 19
2.3.3 Algorithmes d'optimisation locale sous contraintes . . . 23
2.3.4 Algorithmes d'optimisation globale . . . 26
2.4 Approches multiniveaux . . . 31
2.4.1 Vocabulaire et contexte . . . 33
2.4.2 Optimisation multiniveaux de paramètres . . . 34
2.4.3 Optimisation multiniveaux de modèles . . . 37
3 Calculs d'assemblages pour l'optimisation . . . 39
3.1 La méthode LATIN . . . 39
3.1.1 Hypothèses, notations et problème de référence . . . 42
3.1.2 Décomposition de domaine . . . 44

3.1.3 Séparation des difficultés 46
3.1.4 Algorithme LaTIn 48
3.1.5 Lois de comportement des interfaces 50
3.1.6 Aspects pratiques de la résolution et spécificités de la méthode 51
3.1.7 Quelques compléments sur la méthode LATIN 52
3.2 Une méthode de calcul pour les études paramétriques ou l'optimisation 53
3.2.1 Quelques stratégies pour les multiples résolutions . . . 54
3.2.2 Approches numériques pour les multiples résolutions . 58
3.2.3 La stratégie multiparamétrique 59
4 Modèles de substitution 62
4.1 Notations 63
4.2 Mise en œuvre d'un modèle de substitution 63
4.3 Inclusion dans un processus d'optimisation 65
4.4 Méthodes d'échantillonnage de l'espace des paramètres 66
4.4.1 Rappels sur les plans d'expérience usuels 67
4.4.2 Méthodes de remplissage de l'espace (*Space Filling Design*) 69
4.4.3 Bilan des stratégies de tirages 75
4.5 Modèles de substitution 75
4.5.1 Historique des modèles de substitution 76
4.5.2 Régression polynomiale 77
4.5.3 Moindres carrés mobiles 79
4.5.4 Interpolation Cumulative 80
4.5.5 Réseaux de neurones 81
4.5.6 Régression à Vecteur de Support 83
4.6 Validation 85
4.6.1 MSE, RMSE 86
4.6.2 R^2 86
4.6.3 Erreur relative moyenne absolue (RAAE) 86
4.6.4 Erreur relative maximale absolue (RMAE) 87
4.6.5 Quelques erreurs normées 87
4.6.6 Erreurs basées sur l'emploi de la validation croisée . . . 87
4.7 Comparaisons de métamodèles 88
5 Bilan et cadre de l'étude proposée 90

2 La stratégie multiparamétrique au service de l'optimisation d'assemblages 91
1 Introduction 92
2 Mise en œuvre de la stratégie 92
2.1 Aspects pratiques des études paramétriques 92
2.1.1 Première solution 93
2.1.2 Solution retenue, mise en place et développée 93

2.2 Moyens de calcul 95
2.3 Présentation des performances 95
2.3.1 Mesure de performance 96
2.3.2 Influence du type de paramètres 97
2.3.3 Influence du critère d'arrêt 104
2.3.4 Problématique de l'initialisation de la stratégie mutlipa-ramétrique 109
2.3.5 Stratégies d'initialisation 110
2.3.6 Etude de choix d'une stratégie d'initialisation 111
2.3.7 Méthodes de tri des jeux de paramètres 114
2.4 Bilan intermédiaire : 119
3 Calcul des gradients 119
3.1 Méthodes de calcul des gradients 119
3.1.1 Différences finies 120
3.1.2 Approche par variable complexe 124
3.1.3 Méthode directe 125
3.1.4 Méthode adjointe 126
3.1.5 Différenciation automatique 126
3.2 Stratégie multiparamétrique et calcul des gradients 127
3.2.1 Problématique des différences finies appliquées à la stratégie multiparamétrique 127
3.2.2 Recherche du pas des différences finies 128
4 Illustrations des performances 130
5 Bilan sur l'emploi de la stratégie multiparamétrique pour l'optimisation . 132

3 Modèles de substitution et prise en compte des gradients 133
1 Introduction 134
2 Krigeage et RBF sans gradient 134
2.1 Krigeage 134
2.1.1 Élaboration du krigeage 135
2.1.2 Principe de construction 136
2.1.3 Différents types de krigeage 138
2.1.4 Structure de covariance 143
2.1.5 Interpolation 146
2.1.6 Illustration du krigeage 147
2.1.7 Détermination des paramètres 147
2.1.8 Calculs des gradients du krigeage 153
2.1.9 Quelques aspects pratiques de la mise en œuvre du kri-geage 153
2.2 Fonctions de base radiales (RBF) 156
2.2.1 Formulation du modèle 156
2.2.2 Choix de la fonction de base radiale 157
2.2.3 Détermination des paramètres 159

2.2.4 Variance de prédiction des RBF . . . 159
2.2.5 Version étendue des RBF . . . 160
3 Modèles de substitution à gradients . . . 160
3.1 Cokrigeage à gradients . . . 160
3.1.1 Formulation du cokrigeage à gradients . . . 161
3.1.2 Cokrigeage indirect . . . 170
3.1.3 Extension à des dérivées d'ordre supérieur . . . 171
3.2 RBF à gradients . . . 171
3.3 Autres métamodèles à gradients . . . 173
4 Comparaisons de métamodèles . . . 173
4.1 Fonction d'une variable . . . 174
4.1.1 Réponses . . . 174
4.1.2 Intervalles de confiance . . . 176
4.2 Fonction de deux variables . . . 178
4.2.1 Prise en compte de l'anisotropie . . . 178
4.2.2 Prise en compte des gradients . . . 182
4.3 Dimensions supérieures . . . 184
4.4 Bilan . . . 187
5 Métamodèles et stratégie multiparamétrique . . . 188
5.1 Illustration sur un exemple simple . . . 188
5.1.1 Exemple à nombre de tirages fixés . . . 189
5.1.2 Exemple à qualité fixée . . . 190
5.1.3 Montage fretté . . . 190
5.2 Application à des exemples comportant plus de variables de conception . . . 193
6 Bilan de l'élaboration des modèles de substitution . . . 198

4 Mise en œuvre d'une méthode d'optimisation à deux niveaux 201
1 Description de l'approche considérée . . . 202
2 Premier niveau . . . 203
2.1 Construction du modèle de substitution . . . 203
2.2 Optimisation globale sur le métamodèle . . . 203
2.2.1 Méthodes d'enrichissement . . . 204
2.2.2 Mise en œuvre de l'optimisation globale . . . 211
2.2.3 Exemples d'application sur une fonction analytique . . . 212
3 Second niveau . . . 216
4 Application de la procédure d'optimisation à des assemblages . . . 216
4.1 Etude de l'impact d'un rainurage sur une répartition de pression . . . 216
4.1.1 Problème d'optimisation . . . 218
4.1.2 Etude paramétrique . . . 219
4.1.3 Approche sans enrichissement . . . 220
4.1.4 Approche avec enrichissement . . . 220
4.2 Montage fretté . . . 222

4.3 Joint d'accouplement 225
5 Bilan de la stratégie d'optimisation proposée 227

Conclusions et perspectives **229**

Bibliographie **233**

Table des figures

1.1 Exemples d'assemblages de structures 11
1.2 Différents types d'optimisation de forme [DUYSINX 1996] 13
1.3 Quelques exemples de minima globaux et locaux 16
1.4 Exemple de fonction à 2 variables et minima associés avec et sans contrainte d'inégalité 18
1.5 Classification des algorithmes méta-heuristiques en fonction des caractéristiques de fonctionnement [DRÉO 2004] 32
1.6 Décomposition multiniveaux et multi-disciplinaire (schéma de principe adapté de [ENGELS, W. BECKER et A. MORRIS 2004]) 33
1.7 Principe de l'optimisation séquentielle multiniveaux de paramètres, illustration sur 2 niveaux 35
1.8 Principe de l'optimisation itérative multiniveaux de paramètres 36
1.9 Principe de l'optimisation multiniveaux de modèles hiérarchiques, illustration sur 2 niveaux 38
1.10 Principe de l'optimisation multiniveaux de modèles imbriqués 38
1.11 Représentation schématique du comportement de deux stratégies de résolution 40
1.12 Problème de référence 42
1.13 Principe de la décomposition de domaine sans recouvrement en sous-domaines et interfaces 45
1.14 Décomposition prise en compte dans la méthode LATIN et grandeurs aux interfaces 46
1.15 Représentation schématique des itérations de l'algorithme LATIN 48
1.16 Principe et représentation schématique de la stratégie multiparamétrique basée sur la méthode LATIN 61
1.17 Positionnement du métamodèle au sein du contexte de la modélisation du réel issu de [KLEIJNEN et SARGENT 2000] 62
1.18 Représentation schématique d'un modèle de substitution 64
1.19 Représentation schématique du processus d'élaboration d'un modèle de substitution (adaptée de [FORRESTER, SÓBESTER et KEANE 2008]) . . . 65
1.20 Plan factoriel complet 3^3 (27 points) 68
1.21 Exemples de plans fractionnaires issus du plan complet 3^3 68
1.22 Plan composite centré pour 3 paramètres et 2 niveaux 68

1.23 Plan de Box-Behnken pour 3 paramètres 69
1.24 Exemple de 5 sites obtenus par LHS 70
1.25 Quelques exemples de tirages de type *Latin Hypercube Sampling* en dimension 2 pour $n_s = 20$ 71
1.26 Schéma de principe d'un perceptron multi-couche 82

2.1 Présentation synoptique du principe de pilotage de la stratégie multiparamétrique depuis MATLAB 94
2.2 Géométries des cas-tests (**en rouge et en gras**, les paramètres considérés) 98
2.3 Efforts résultants à la paroi 101
2.4 Temps de calcul et gains pour les deux échantillonnages et pour les 4 cas-tests 102
2.5 Efforts résultants à la paroi 105
2.6 Ecart sur la résultante 106
2.7 Courbes de convergence de la stratégie multiparamétrique du cas-test « CSMA » pour 20 jeux de paramètres et 2 critères d'arrêt de l'algorithme LATIN 107
2.8 Temps CPU et gain de la stratégie multiparamétrique en fonction du critère d'arrêt pour les 4 exemples d'assemblages 108
2.9 Principe des stratégies d'initialisation de l'algorithme LATIN pour obtenir la solution $\mathbf{s}^k$ 111
2.10 Temps CPU et gains obtenus pour 5 stratégies d'initialisation (cas-test `CSMA`, $\eta = 10^{-4}$, 19 valeur de n_s échantillonnés 30 fois chacune par LHS) 112
2.11 Evolutions des distances intersites et dépendance du gain de la stratégie multiparamétrique à la moyenne des distance intersites (cas-test `CSMA`, $\eta = 10^{-4}$, 19 valeur de n_s échantillonnés 30 fois chacune par LHS, initialisation `para L2`) 114
2.12 Valeur de l'effort à la paroi en fonction des paramètres pour le cas-test `CSMA` à 2 paramètres (figure obtenue par une grille régulière 20×20 dans l'espace des paramètres) 117
2.13 Temps CPU et gain de la stratégie multiparamétrique pour différentes méthodes de tri des paramètres (cas-test `CSMA`, 2 paramètres, $\eta = 10^{-4}$, 19 valeurs de n_s échantillonnées 30 fois chacune par LHS, initialisation `para L2`) 117
2.14 Temps CPU et gain de la stratégie multiparamétrique pour différentes méthodes de tri des paramètres et pour différents jeux de paramètres de référence $\mathbf{x}_{ref}$ (cas-test `CSMA`, 2 paramètres, $\eta = 10^{-4}$, 19 valeurs de n_s échantillonnées 30 fois chacune par LHS, initialisation `para L2`) 118
2.15 Gradients de l'effort à la paroi en fonction du coefficient de frottement à l'interface pour $\mu = 0.1$ (cas-test CSMA) 129
2.16 Gradients de l'effort à la paroi en fonction du coefficient de frottement à l'interface pour $\mu = 0.6$ (cas-test CSMA) 130
2.17 Géométrie du cas-test CSMAb 131

3.1 Intégration d'un métamodèle pour approcher une réponse coûteuse 134
3.2 Fonctions de corrélation Gaussienne et Matérn pour 3 valeurs de la portée ℓ 145
3.3 Approximations et variances obtenues par krigeage ordinaire pour 3 paramètres de corrélation (fonction de corrélation gaussienne) 148
3.4 Approximations et variances obtenues par krigeage ordinaire pour 3 tirages comportant 3, 6 et 10 points (fonction de corrélation gaussienne, paramètre $\ell = 1$) 148
3.5 Approximations par krigeage ordinaire (G. : corrélation gaussienne, M. : corrélation Matérn ($\nu = 3/2$)) 152
3.6 Erreur d'approximation ($\mathtt{Q}_3$ voir page 87), erreur de validation croisée (CV) et -log-vraisemblance 153
3.7 Conditionnement de la matrice de corrélation pour 4 fonctions de corrélation (échantillonnages réguliers de l'espace $[0, 15]$ et sans normalisation des sites) 155
3.8 Exemples de fonctions de base radiales pour 3 valeurs du paramètre ℓ . . 158
3.9 Exemples de fonctions de corrélations Gaussienne et Matérn (valeur de la fonction, dérivées premières et dérivées secondes pour $\ell = 0.8$) 168
3.10 Approximations avec et sans gradients d'une fonction d'une variable (échantillonnage LHS, $n_s = 6$) 175
3.11 Dérivées des approximations (échantillonnage LHS, $n_s = 6$) 176
3.12 Approximations de type krigeage et intervalles de confiance (échantillonnage LHS, $n_s = 6$) 177
3.13 Approximations de type RBF et intervalles de confiance (échantillonnage LHS, $n_s = 6$) 177
3.14 Fonctions test 179
3.15 Métamodèles isotropes 180
3.16 Métamodèles anisotropes 181
3.17 Fonction *Six-Hump Camel Back* approchée par métamodèles de krigeage avec et sans gradients (échantillonnage IHS, $n_s = 20$) 182
3.18 Fonction *Schwefel* approchée par métamodèles de krigeage avec et sans gradients (échantillonnage IHS, $n_s = 40$) 182
3.19 Qualité d'approximation de la fonction Rosenbrock à 5 variables obtenue avec 50 exécutions de chaque tirage 185
3.20 Qualité d'approximation de la fonction Rosenbrock à 10 variables obtenue avec 50 exécutions de chaque tirage 185
3.21 Qualité d'approximation de la fonction Schwefel à 5 variables obtenue avec 50 exécutions de chaque tirage 186
3.22 Qualité d'approximation de la fonction *Schwefel* à 10 variables obtenue avec 50 exécutions de chaque tirage 186
3.23 Rappels du cas-test CSMA 189
3.24 Surfaces de réponse des modèles de substitution de type krigeage obtenues à partir de 15 appels au solveur 189
3.25 Géométrie du problème de frettage 191

3.26 Géométrie nominale et avec défaut de l'arbre 192
3.27 Surface de réponse de référence du cas-test de frettage (contrainte de Von Mises maximale dans l'assemblage) 192
3.28 Approximations par krigeage et cokrigeage ordinaire obtenues à partir de 30 et 10 points (échantillonnage LHS) 193
3.29 Géométrie du cas-test CSMAb 195
3.30 Approximations fournies pour le cas-test CSMAb à 3 paramètres 195
3.31 Approximations fournies pour le cas-test CSMAb à 4 paramètres 196
3.32 Temps de calcul et gains associés aux approximations dont les qualités sont données en figure 3.30 pour l'exemple CSMAb à 3 paramètres 197
3.33 Temps de calcul et gains associés aux approximations dont les qualités sont données en figure 3.31 pour l'exemple CSMAb à 4 paramètres 197
3.34 Evolution de la qualité d'approximation en fonction du temps CPU pour les cas-tests CSMAb à 3 et 4 paramètres 198

4.1 Schéma synoptique de l'optimisation à deux niveaux de modèles 202
4.2 Fonction d'exemple d'une variable et approximation par krigeage ordinaire 205
4.3 Enrichissement par l'optimum et convergence locale (ajout d'un seul nouveau point) 206
4.4 Enrichissement par la variance maximale : approximation et variance du krigeage ordinaire (ajout d'un seul nouveau point) à comparer avec la figure 4.2 207
4.5 Enrichissement successif d'un métamodèle de krigeage ordinaire 210
4.6 Fonction *Six-Hump Camel Back* 214
4.7 Allure des surfaces de réponses de l'amélioration espérée et de l'approximation fournie par krigeage ordinaire aux itérations 1, 5 et 10 de l'algorithme EGO 214
4.8 Allure des surfaces de réponses de l'amélioration espérée et de l'approximation fournie par cokrigeage ordinaire à gradients aux itérations 1, 5 et 10 de l'algorithme EGO 215
4.9 Géométrie éclatée du cas-test 217
4.10 Géométrie de la rainure et paramètres 217
4.11 Exemple de champ de pression à l'interface $\Omega_9 - \Omega_{10}$ pour $X = 72.9\,\text{mm}$, $L = 24.3\,\text{mm}$ et $h = 0.5\,\text{mm}$ et position de la rainure 218
4.12 Composante σ_{yy} du champ des contraintes 218
4.13 Surface de réponse de la fonction objectif considérée obtenue à $\eta = 10^{-4}$ 219
4.14 Exemple d'assemblage démontable par frette conique 223
4.15 Problème axi-symétrique 2D 223
4.16 Joint d'accouplement 226
4.17 Contrainte de cisaillement maxi dans la vis la plus chargée de l'assemblage 227

Liste des tableaux

1.1 Les grands développements autour de la méthode LATIN 41
1.2 Articles de comparaisons et conclusions 89

2.1 Spécifications des 4 cas-tests présentés dans la figure 2.2 99
2.2 Quelques caractéristiques des calculs sur les 4 assemblages (critère d'arrêt LATIN $\eta = 10^{-4}$) 103
2.3 Stratégies de tri des tirages 116
2.4 4 points de référence considérés 116
2.5 Coefficients de calcul des différences finies 121
2.6 Caractéristiques des calculs avec et sans calcul des gradients pour les cas-tests CSMA et CSMAb avec 3 et 4 paramètres 132

3.1 Composantes des approximations fournies par les 3 modèles de krigeage . 143
3.2 Quelques fonctions de corrélation 144
3.3 Approximations et variances du krigeage exprimées à l'aide des fonctions de corrélation 146
3.4 Quelques fonctions de base radiales 157
3.5 Quelques références concernant l'emploi du cokrigeage à gradients 162
3.6 Abréviations employées 174
3.7 Critères de qualité globaux des approximations de la figure 3.10 (la signification des critères peut être trouvée page 85) 175
3.8 Fonctions test employées pour évaluer la qualité d'approximation des modèles de substitution 178
3.9 Caractéristiques des métamodèles isotropes et anisotropes 181
3.10 Caractéristiques des métamodèles avec et sans gradients en 2D sur les fonctions test *Six-Hump Camel Back* (S.H.C.B.) et *Schwefel* construits avec des tirages IHS comportant respectivement 20 et 40 points 183
3.11 Caractéristiques des métamodèles avec et sans gradients en 2D sur la fonction test *Six-Hump Camel Back* construits avec un tirage IHS comportant respectivement 30 points (en comparaison des résultats obtenus avec 20 points dans le tableau 3.10) 184
3.12 Performances mécaniques et qualité d'approximation des métamodèles de krigeage et cokrigeage construits 190

3.13 Krigeage et cokrigeage construits à partir de plans factoriels complets . . 191
3.14 Caractéristques des métamodèles et des calculs mécaniques du cas-test de frettage 194

4.1 Evolution des grandeurs lors de l'optimisation par EGO basée sur le krigeage ordinaire 213
4.2 Evolution des grandeurs lors de l'optimisation par EGO basée sur le cokrigeage ordinaire à gradients 213
4.3 Caractéristiques du calcul mécanique permettant d'obtenir la surface de réponse figure 4.13 pour une grille 25×25 220
4.4 Mise en défaut de l'approche sans enrichissement sur le cas-test de la rainure 221
4.5 Résultats du processus d'enrichissement sur le cas-test de la rainure . . . 221
4.6 Résultats du processus d'optimisation sur la frette comportant 5 paramètres 224

Introduction

Alors qu'au sein des bureaux d'études les démarches de conception s'orientent largement autour du dimensionnement de pièces isolées, la problématique de l'assemblage des structures n'est aujourd'hui résolue que de manière partielle. En effet, la réunion de deux pièces d'un assemblage fait apparaître de nombreuses incertitudes liées par exemple à une mauvaise connaissance du comportement des pièces et des interfaces entre les pièces pendant toute la durée de vie de l'assemblage. On pourra citer par exemple la problématique liée à l'assemblage de pièces en matériau composite par liaisons boulonnées dans le domaine de l'aéronautique, où au comportement complexe du matériau en lui même, s'ajoute la difficulté de comprendre les mécanismes permettant de concevoir et de dimensionner une liaison boulonnée adaptée. À une autre échelle, dans le contexte de l'assemblage de grandes structures flexibles (la flexibilité est due à l'élancement des pièces considérées), les phases de conception et de dimensionnement nécessitent d'être capable de comprendre et de maîtriser les phénomènes mécaniques survenant au niveau des liaisons. Ces connaissances *a priori* permettent de s'assurer de la faisabilité, de la mise en œuvre et de la durée de vie de l'assemblage.

L'élaboration d'essais expérimentaux sur des assemblages est dans un premier temps inappropriée en raison des coûts financier et temporel, de la difficulté d'instrumentation ainsi que du post-traitement des résultats qui nécessitent une importante connaissance *a priori* du comportement de l'assemblage. La simulation numérique, basée par exemple sur la méthode des éléments finis, a permis de fournir une aide non négligeable à la conception d'assemblages de structures. Malheureusement, les phénomènes à prendre en compte sont fortement non-linéaires du fait de la présence de frottement et/ou de contact entre les pièces. Le temps de calcul qui en résulte est alors extrêmement important et la précision des calculs est souvent mal connue.

C'est en raison de ces incertitudes que la conception d'assemblages nécessite à l'heure actuelle l'usage de coefficients sécurité importants afin de prendre en compte tous les problèmes pouvant survenir lors de la vie de l'assemblage. Malheureusement, ces coefficients conduisent à un sur-dimensionnement de l'assemblage qui engendre un sur-coût. Une meilleure prédiction du comportement de l'assemblage et l'usage de méthodes de conception adaptées à ce type de problèmes pourraient conduire à une augmentation des performances obtenues par une diminution non négligeable des masses.

Parallèlement à cette problématique de simulation numérique, l'emploi de stratégies

d'optimisation apparaît comme incontournable. Bien que les optimisations paramétrique et de forme soient des outils de plus en plus employés pour la conception de pièces (géométrie, caractéristiques matériaux...) et que l'optimisation multi-disciplinaire et l'optimisation topologique émergent, l'optimisation d'assemblages de pièces est relativement peu employée. Au vu des verrous énoncés précédemment, les principales causes de cette faible utilisation peuvent être identifiées.

- La conception d'assemblages peut nécessiter l'intervention de plusieurs services du bureau d'études. Chaque service ayant ses objectifs, ses méthodes, ses disciplines, leur mise en commun et le traitement conjoint du problème peut s'avérer particulièrement difficile.
- Les phénomènes survenant lors de la mise en œuvre d'un modèle de calcul d'assemblages (tels que les différents types de comportement régissant le contact entre pièces) étant pour la plupart fortement non-linéaires, ils nécessitent des stratégies de résolution mécanique très évoluées qui ne sont pas toujours maîtrisées par les utilisateurs.
- Le temps de calcul nécessaire à la résolution d'un problème d'assemblage de structures associé à un jeu de paramètres de conception est très important. Or, les stratégies d'optimisation classiques nécessitent un nombre important de réponses du modèle mécanique pour converger vers le minimum global. Ainsi, ces stratégies d'optimisation sont quasi-systématiquement inabordables.

C'est au sein du projet ANR OMD2 (ANR-08-COSI-007-10, projet débuté en 2009 pour 36 mois) que s'insèrent les travaux proposés ici. Ce dernier fait suite au projet exploratoire ANR OMD (ANR-05-RNTL-009, OMD : Optimisation MultiDisciplinaire, projet débuté en 2005 pour 36 mois) visant à proposer et à mettre en place des méthodologies et démonstrateurs afin d'être capable de résoudre des problèmes d'optimisation multidisciplinaire. Le projet OMD2 (Optimisation MultiDisciplinaire Distribuée) porté par Renault S.A. a principalement pour but de mettre en place une plateforme capable de mener à bien des problèmes d'optimisation multidisciplinaire de grande dimension nécessitant l'emploi de démarches de résolution numérique mécanique adaptées et robustes. La problématique HPC (High Performance Computing) est ainsi au cœur de ce second projet. Le projet OMD2 est scindé en *workpackages* et les différentes thématiques du projet (autour du calcul fluide, de structures, de développements informatiques, de développements d'outils d'optimisation...) font l'objet d'un travail collaboratif entre plusieurs laboratoires (INRIA, le laboratoire MAS de l'Ecole Centrale Paris, le département 3MI de l'Ecole Nationale des Mines de Saint-Etienne, le laboratoire Roberval de l'Université Technologique de Compiègne et le LMT-Cachan) et PME (CD-Adapco France, ActiveEon, Sirehna). Le LMT-Cachan a proposé au sein de ce projet l'emploi de stratégies adaptées à la résolution numérique de problèmes d'assemblages et leur utilisation pour construire des approximations de fonctions objectifs à moindre coût.

Plus récemment, le démarrage du projet CARAB (Conception Avancée Robuste pour les Assemblages Boulonnés, projet débuté en 2012) porté par la SNECMA (et auquel participe le LMT-Cachan) montre l'importance que prend cette question du calcul d'assemblages dans l'industrie. Une meilleure connaissance et maîtrise des assemblages, l'intégration et le déploiement de stratégies de calcul dédiées et de démarches adaptées à la conception d'assemblages semblent aujourd'hui incontournables dans les bureaux d'études.

Parmi ces démarches, l'adaptation de stratégies d'optimisation à ce type de problèmes apparaît comme un élément prépondérant de la conception et du dimensionnement de systèmes mécaniques.

Cette thèse se positionne donc à l'interface des thématiques relatives à l'optimisation et au calcul numérique d'assemblages. Elle tente de mettre en place une stratégie dédiée à l'optimisation d'assemblages avec pour objectif de rendre ce type d'optimisation abordable en termes de coût de calcul. La mise en œuvre d'une telle démarche ne peut se concevoir qu'en étant capable de résoudre à coût limité les problèmes d'assemblages.

Ces travaux s'insèrent dans le contexte de l'optimisation globale, c'est-à-dire que l'on cherche à localiser le minimum global de la fonction objectif ainsi que les paramètres associés tout en étant capable de s'affranchir d'éventuels minima locaux (par opposition à une optimisation locale qui cherche à trouver un minimum situé dans le voisinage du jeu de paramètres employé pour initialiser la recherche ; le minimum atteint dans le cadre de l'optimisation locale est un optimum local qui peut être global) et de contraintes. Afin de parvenir à cet objectif, il est possible de recourir à l'emploi d'algorithmes de recherche globale « classiques » (tels que, par exemple, les algorithmes génétiques ou les méta-heuristiques basées sur le comportement des essaims particulaires). Cependant, ce type d'algorithmes requérant un nombre extrêmement important d'évaluations de la fonction objectif pour mener à bien le processus de recherche de minimum global, il n'est clairement pas envisageable de les utiliser directement avec un modèle capable de résoudre le problème d'assemblage. C'est pourquoi, une approche à deux niveaux de modèles sera proposée. Elle forme avec tous les outils nécessaires à son fonctionnement, la structure de ces travaux.

Les deux niveaux qui composent la stratégie d'optimisation retenue se basent tous deux sur l'emploi d'un modèle sur lequel une optimisation est réalisée.

1. Le premier niveau se base sur l'emploi d'un *modèle de substitution* ou *métamodèle* construit à partir d'évaluations et, selon le cas de figure, de gradients d'une fonction objectif fournis par le modèle mécanique. Ces informations sont obtenues en un nombre limité de jeux de paramètres fournis par une stratégie de tirage classique. Le métamodèle est capable de fournir une très bonne qualité d'approximation globalement sur l'espace de conception. Il peut cependant nécessiter l'usage d'un processus d'enrichissement permettant l'ajout « intelligent » de nouveaux jeux de paramètres

(et des réponses et gradients associés). Cet enrichissement permet d'assurer une localisation plus fine du minimum global. La stratégie d'**optimisation globale** assurant la localisation des zones potentielles où se trouve le minimum global est réalisée avec temps de calcul très faible. En effet, les évaluations obtenues grâce au modèle de substitution sont extrêmement peu coûteuses. À ce niveau de la stratégie d'optimisation, les seules opérations coûteuses en termes de temps de calcul sont donc celles liées aux appels au solveur mécanique lors de la construction du métamodèle et de son enrichissement.

2. Le second niveau consiste, quant à lui, en une **optimisation locale** réalisée directement avec le modèle mécanique. Celle-ci est rendue viable grâce à l'initialisation de l'algorithme de recherche locale : ce dernier est initialisé à partir des jeux de paramètres issus des zones potentielles retenues sur le métamodèle lors de l'optimisation globale du premier niveau. Cette stratégie permet l'obtention précise du minimum global et du jeu de paramètres associé. Dans certains cas, cette seconde optimisation peut s'avérer cependant inutile en raison de la présence de la phase d'enrichissement au premier niveau.

La stratégie complète fait alors apparaître des appels au modèles mécanique coûteux en temps de calcul :

- lors du calcul des réponses et gradients fournis par le modèle mécanique nécessaires à la construction du métamodèle dans le premier niveau ;
- lors de la phase d'optimisation locale du second niveau.

Cependant, en fonction du nombre de jeux de paramètres choisis initialement, du type de modèles de substitution retenu et de divers paramètres de réglage des phases d'enrichissement et d'optimisation locale et globale, on remarque que la répartition des temps de calcul peut varier entre les niveaux.

Dans ce contexte de réduction des temps de calcul inhérents aux calculs d'assemblages, il a été nécessaire d'employer une stratégie dédiée à ce type de résolution. Pour ce faire, la **stratégie multiparamétrique** a été utilisée. Elle est basée sur l'approche **LaTIn** (*Large Time Increment method*) introduite par [LADEVÈZE 1985 ; LADEVÈZE 1999]. L'approche LaTIn consiste à considérer que le problème est composé d'*interfaces* et de *sous-domaines* ayant chacun leur comportement propre. Dans le contexte des assemblages élastiques, cette décomposition est relativement naturelle. En effet, dans ce cadre, les pièces peuvent être considérées comme sous-domaines et les zones de contact entre les pièces comme interfaces. En utilisant une méthode de décomposition de domaine mixte, le problème peut alors être construit en regroupant, d'une part, les équations liées aux interfaces et, d'autre part, celles liées aux sous-domaines. La résolution successive de ces deux groupes d'équations et l'utilisation de directions de recherche adaptées assure la convergence de la méthode vers la solution. La méthode LaTIn présente la particularité de fournir à chaque

itération une solution sur l'ensemble de l'espace et du temps. Chaque itération assure alors un enrichissement de la solution atteinte à convergence (la convergence est pilotée par l'emploi d'un critère d'arrêt basé sur un indicateur d'erreur global spécifique à la méthode LaTIn). Par ailleurs, dans le cadre de l'étude proposée ici, les sous-domaines seront considérés comme élastiques linéaires sous l'hypothèse des petites perturbations. Cette hypothèse permet alors de considérer que la solution est intégralement définie par les grandeurs aux interfaces (efforts et déplacements).

C'est en se basant sur ces particularités de l'approche LaTIn que la stratégie multiparamétrique a été envisagée. Lorsque l'on souhaite obtenir la solution au problème d'assemblage associée à un jeu de paramètres donné, la stratégie multiparamétrique consiste en une réinitialisation de l'algorithme LaTIn par une solution (définie par les grandeurs d'interface) précédemment convergée et associée à un jeu de paramètres donné. Un choix judicieux de la solution utilisée pour réinitialiser l'algorithme entraîne une importante réduction du nombre d'itérations et par conséquent du temps de calcul. Cette technique introduite par [BOUCARD et LADEVÈZE 1999] et employée par exemple dans [BOUCARD 2001 ; BOUCARD et CHAMPANEY 2003] permet dans un contexte d'optimisation ou d'étude paramétrique d'obtenir à coût réduit des dérivées de la fonction objectif par rapport aux paramètres en utilisant une méthode par différences finies.

Ainsi, la fonction et ses gradients étant disponibles à moindre coût, l'emploi d'un **métamodèle** capable de prendre en compte les gradients s'est avéré incontournable. Bien qu'il existe de nombreux types de métamodèles, notre choix s'est porté sur deux types de métamodèles capables de fournir une approximation de qualité : le *krigeage* et les *RBF* (*Radial Basis Functions*). Ces deux types de modèles de substitution sont largement employés dans le cadre de l'optimisation et possèdent des formulations permettant la prise en compte des gradients. Le métamodèle de krigeage, introduit par [KRIGE 1951b ; MATHERON 1963b ; SACKS, SCHILLER et WELCH 1989] est aujourd'hui un des plus utilisés dans le contexte de l'optimisation. Sa renommée est en partie due la toolbox DACE [1] (*Design and Analysis of Computer Experiments*) [LOPHAVEN, NIELSEN et SØNDERGAARD 2002] ainsi qu'à son déploiement dans de nombreux logiciels tels que dans

- le logiciel-langage statistique R [2] (avec la librairie DiceKriging développée dans le cadre du consortium DICE [3] piloté par le département 3MI de l'Ecole Nationale Supérieure des Mines de Saint-Etienne) ;
- la toolbox STK [4] (MATLAB) développée E. Vazquez et J. Bect ;
- la toolbox DACE [5] (Scilab) développée par Y. Collette et H.B. Nielsen ;
- la toolbox SUMO [6] (MATLAB, Java, C et C++) développée par le SUMO Lab de

1. `http://www2.imm.dtu.dk/~hbni/dace/`
2. `http://cran.r-project.org/`
3. `http://dice.emse.fr/`
4. `http://sourceforge.net/projects/kriging/`
5. `http://atoms.scilab.org/toolboxes/dace_scilab`
6. `http://www.sumo.intec.ugent.be/SUMO`

l'Université de Gand (*IBCN research group*) ;

- la toolbox GPML [7] (MATLAB) développée par C.E. Rasmussen et C.K.I. Williams ;
- la toolbox krisp [8] (Scilab) développée par J. Janusevskis...

Avec une mise en œuvre plus simple, le modèle de substitution basé sur les fonctions de bases radiales [HARDY 1971] assure également une bonne qualité d'approximation. En s'appuyant sur des travaux antérieurs, ces deux modèles de substitution ont été, lors de cette thèse, développés et étudiés dans leur version enrichie avec les gradients de la fonction objectifs. Ils forment aujourd'hui le coeur de la toolbox GRENAT (*GRadient ENhanced Approximation Toolbox*) développée au sein du LMT-Cachan.

Ce manuscrit présente en détail la démarche que nous venons brièvement d'évoquer. Les différents outils employés seront détaillés et leurs performances seront illustrées sur des cas-tests analytiques et mécaniques. Ce document est structuré comme suit :

- le **premier chapitre** vise à présenter un tour d'horizon des outils nécessaires à l'optimisation d'assemblages. Ils seront successivement introduits et développés. Ainsi, après une partie relative à l'optimisation où le formalisme, les différentes problématiques et stratégies de résolution concernant ce type de problèmes seront présentées, la méthode de calcul d'assemblage basée sur la méthode LaTIn et la stratégie multiparamétrique seront rappelées. Enfin, ce chapitre introduira les modèles de substitutions de manière générale ainsi que les problématiques liées à leur mise en œuvre. Quelques métamodèles courants seront développés de manière succincte.
- le **deuxième chapitre** vise à présenter de manière détaillée l'emploi de la stratégie multiparamétrique pour calculer la solution du problème d'assemblage dans un contexte d'optimisation. Les performances de la méthode seront illustrées sur un ensemble de cas-tests faisant intervenir les divers comportements aux interfaces présents dans ce contexte d'assemblage de structures. Après un état de l'art des méthodes de calcul des gradients, la stratégie multiparamétrique sera employée dans ce but. Dans un dernier temps, la stratégie multiparamétrique sera mise en œuvre sur un cas-test 2D.
- le **troisième chapitre** s'attelle à la mise en œuvre de modèles de substitution capables de prendre en compte les gradients de la fonction objectif. Les problématiques liées à l'échantillonnage de l'espace de conception, à la détermination des paramètres internes des métamodèles et à la validation de ces métamodèles seront abordées. Les métamodèles de type krigeage et RBF seront présentés en détail dans leur version avec et sans gradients. Un ensemble de résultats illustrera en détail les performances

7. `http://www.gaussianprocess.org/gpml/code/matlab/doc/`
8. `http://atoms.scilab.org/toolboxes/krisp`

en termes de qualité d'approximation de ces métamodèles en employant des fonctions test analytiques ayant des comportements plus ou moins non-linéaires. Enfin, la stratégie multiparamétrique sera utilisée pour obtenir les réponses et gradients de modèles mécaniques nécessaires à la construction de ces modèles de substitution. Une étude illustrant les performances de construction des métamodèles en termes de coût de calcul et de qualité d'approximation sera par ailleurs proposée.

- le **quatrième chapitre** visera à réunir les méthodes introduites auparavant. La stratégie d'optimisation multiniveaux sera développée en détail puis appliquée sur différents cas-tests mécaniques de complexité croissante. Dans ce chapitre, les problématiques d'optimisation globale et locale appliquées respectivement sur le modèle de substitution et le solveur mécanique seront présentées. L'emploi de cas-tests d'assemblages en 2D et 3D permettra d'illustrer les performances de la stratégie d'optimisation multiniveaux proposée.

Chapitre 1

Outils au service de l'optimisation d'assemblages

Ce premier chapitre propose de préciser le contexte de l'optimisation d'assemblages. Tout d'abord, le cadre de l'optimisation sera présenté ainsi que les méthodes pour résoudre ce type de problèmes. Ensuite, le cadre des méthodes de résolution de problèmes d'assemblages sera proposé. En particulier, la stratégie multiparamétrique et la méthode LATIN *sur laquelle elle est basée seront détaillées. Dans un dernier temps, les méthodes d'approximation de type modèle de substitution seront présentées ainsi que le contexte de leur mise en œuvre.*

Sommaire

1 Introduction 10
2 Optimisation 12
2.1 Différents types d'optimisation et de stratégies 12
2.2 Formulation d'un problème d'optimisation paramétrique 14
2.3 Algorithmes d'optimisation 19
2.4 Approches multiniveaux 31
3 Calculs d'assemblages pour l'optimisation 39
3.1 La méthode LATIN 39
3.2 Une méthode de calcul pour les études paramétriques ou l'optimisation 53
4 Modèles de substitution 62
4.1 Notations 63
4.2 Mise en œuvre d'un modèle de substitution 63
4.3 Inclusion dans un processus d'optimisation 65
4.4 Méthodes d'échantillonnage de l'espace des paramètres 66
4.5 Modèles de substitution 75
4.6 Validation 85
4.7 Comparaisons de métamodèles 88
5 Bilan et cadre de l'étude proposée 90

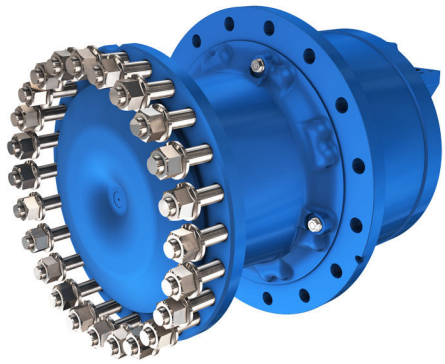
1 Introduction

À l'heure actuelle, alors que de nombreux outils et méthodes d'optimisation font partie des processus de conception, la problématique de leur application au dimensionnement d'assemblages reste peu étudiée. En effet, les problématiques complexes relatives à la résolution mécanique des problèmes d'assemblages nécessitent l'emploi de méthodologies de résolution de haute technicité. De nombreux travaux se sont penchés sur cette question mais les temps de calcul nécessaires à la résolution de ce type de problèmes n'en reste pas moins importants. C'est dans ce contexte qu'il est nécessaire de mettre en place des stratégies dédiées à :

- résoudre les problèmes d'assemblages tout en étant capable de prendre en compte les fortes non-linéarités présentes dans ce type de problèmes.
- mener à bien, sur ce type de problèmes, une optimisation globale et robuste en un temps de calcul réduit et maîtrisé.

De manière générale, le terme **assemblage** désigne une très grande famille d'opérations et techniques visant à réunir différentes pièces afin que celles-ci forment, à l'état final de cette action, un ensemble « rigide » (sans mobilité des pièces les unes par rapport aux autres). Leur mise en œuvre fait intervenir de nombreux phénomènes tels que, par exemple, des phénomènes de frottement, de contact, des phénomènes liés au matériaux (élasticité, plasticité...). Par ailleurs, les procédés peuvent nécessiter des chargements variés (efforts, déplacements, sollicitations thermiques...). Parmi les très nombreuses techniques d'assemblage, on peut citer par exemple l'assemblage de deux pièces à l'aide de vis et l'assemblage par frettage. Ces deux techniques exploitent les phénomènes de frottement et de contact entre les pièces pour assurer la tenue de l'assemblage. Dans les travaux présentés ici, on ne s'intéressera pas aux opérations relatives à la phase d'assemblage des pièces telles que, par exemple, les phases d'enlèvement de matières (perçage...). On considérera l'assemblage de pièces dont les formes finales ont été atteintes.

Afin de concevoir, dimensionner et valider les solutions retenues pour assembler des structures, une démarche basée sur l'emploi de la simulation numérique et de campagnes d'essais expérimentales peut être employée. La réalisation d'essais expérimentaux pouvant s'avérer particulièrement longue et financièrement coûteuse, la simulation numérique est de plus en plus utilisée voire même remplace la phase expérimentale (on parle de « *virtual testing* »). Dans le contexte de la simulation d'assemblages, elle nécessite la mise en œuvre de stratégies extrêmement spécifiques capables de tenir compte des nombreux phénomènes physiques intervenant au sein d'un assemblage aussi bien dans sa phase de mise en œuvre que lors de sa phase d'utilisation. Des phénomènes tels que le **contact** et/ou le **frottement** entre pièces ne peuvent en particulier pas être ignorés : dans le cas d'une liaison boulonnée, la présence éventuelle de **jeu** entre la vis et son logement et le maintien par adhérence des entités assemblées nécessitent une attention particulière. La non prise en compte de ces phénomènes peut engendrer de gros dysfonctionnements (les vis pourraient travailler en cisaillement, ce à quoi elles ne sont *a priori* pas adaptées).



(a) Dispositif d'assemblage par adhérence par liaison boulonnée d'un moteur-roue Poclain[1]



(b) Dispositif d'assemblage par adhérence Haimer[2] à faible conicité d'un porte-outil sur un mandrin de machine outil

FIGURE 1.1 – Exemples d'assemblages de structures

Par ailleurs, le frottement ou le contact sont des phénomènes au comportement extrêmement non-linéaire qui nécessitent des méthodes numériques adaptées pour pouvoir être simulés. Dans ce contexte et afin d'obtenir une représentativité suffisante des phénomènes survenant lors d'un assemblage, le temps nécessaire à la réalisation de la simulation numérique peut être très important. Dans le cadre des travaux proposés ici, on s'intéresse au déploiement de l'optimisation et à son application sur des assemblages de structures. L'optimisation vise à trouver la « meilleure » solution au sens d'un critère tout en respectant ou non une ou plusieurs contrainte(s) (cf. partie 2.2.1 pour plus de détails sur la formulation d'un problème d'optimisation). De nombreuses stratégies et techniques d'optimisation existent (cf. partie 2.3), elles nécessitent cependant un nombre très important de simulations numériques pour parvenir à trouver cette « meilleure » solution. Le temps de calcul de chaque simulation pouvant être très important, l'utilisation directe de techniques d'optimisation « classiques » n'est pas viable dans ce cas.

On propose alors d'utiliser un modèle mathématique pour obtenir une approximation de la réponse de la simulation numérique. Cette approximation pourra alors fournir à coût négligeable (par rapport au coût de calcul nécessaire à une simulation numérique) une réponse associée à un jeu de paramètres. La stratégie d'optimisation se basera en partie sur cette approximation pour localiser la « meilleure » solution approchée. Dans un second temps, une recherche précise de la solution pourra être réalisée à l'aide de la simulation numérique d'assemblage.

Ce chapitre consistera alors à introduire et rappeler les différents principes relatifs aux outils nécessaires à la mise en œuvre de l'approche d'optimisation d'assemblages. Il s'articulera autour de trois axes : l'optimisation, la résolution d'un problème mécanique d'assemblage pour un ensemble de jeux de paramètres et les modèles de substitution.

1. `http://www.poclain-hydraulics.com/`
2. `http://www.haimer.biz/`

2 Optimisation

2.1 Différents types d'optimisation et de stratégies

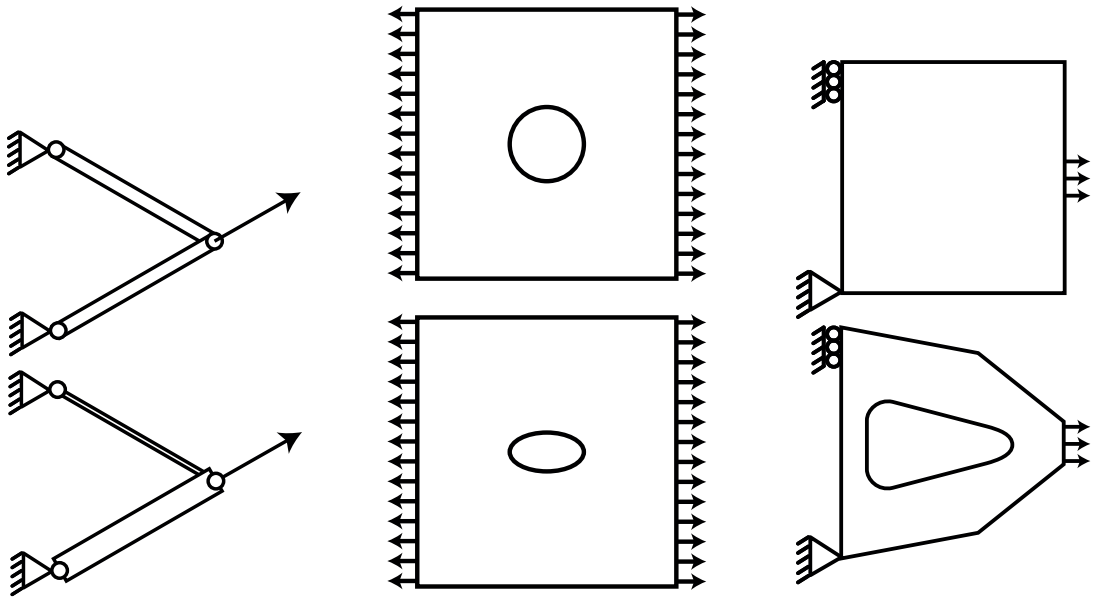
Dans le contexte de la conception mécanique, on peut distinguer différents types d'optimisation et de stratégies spécifiques destinés à trouver la « meilleure » solution. Parmi ces différents types d'optimisation, on peut distinguer deux grandes familles : l'optimisation de forme et l'optimisation paramétrique.

2.1.1 Optimisation de forme

L'optimisation de forme vise à trouver la meilleure forme d'une structure sous sollicitations mécaniques. L'objectif de la stratégie consiste alors à minimiser ou maximiser un (des) critère(s). Pour cela, elle va chercher la solution qui respecte aux mieux le (les) critère(s) parmi un ensemble de solutions admissibles. Ces solutions admissibles doivent respecter les contraintes inhérentes au problème (telles que la prise en compte de conditions aux limites mécaniques et les liaisons qui y sont associées) et/ou les contraintes fixées par le contexte dans lequel se place le problème (telles que des coûts, des critères de tenue mécanique...). L'évaluation du (des) critère(s) de chaque solution admissible envisagée est obtenue grâce à un modèle mécanique fournissant pour chaque solution admissible envisagée (c'est-à-dire pour chaque forme envisagée) la réponse mécanique de la structure aux sollicitations. L'optimisation de forme peut se décomposer en trois types de problèmes [ALLAIRE 2007] : l'optimisation de formes paramétrique, géométrique ou topologique.

■ **L'optimisation de forme paramétrique :** Cette approche, aussi connue sous le terme **dimensionnnement automatique des structures** [DUYSINX 1996], consiste à considérer les formes régies par un ensemble de paramètres tels que des paramètres dimensionnels (épaisseur, dimensions...). Ce type d'optimisation peut rendre extrêmement complexe la gestion des solutions admissibles. En effet, certains jeux de paramètres (que l'on choisirait *a priori* comme variant de façon continue) pourrait conduire à des solutions géométriques impossibles (par exemple des zones à épaisseur nulle). Dans ces cas, la bonne définition du problème d'optimisation de forme impose l'écriture de fonctions contraintes assurant la bonne définition des solutions admissibles. Ces fonctions contraintes consistent en l'écriture de relations d'égalités ou d'inégalités faisant intervenir les paramètres du problème (cf. partie 2.2.1 pour plus de détails concernant la formulation d'un problème d'optimisation).

■ **L'optimisation de forme géométrique :** Cette approche consiste à définir une forme initiale et les différentes solutions admissibles sont obtenues par modification de la position des frontières de la forme sans en modifier la topologie (ce qui pourrait survenir dans le cadre de l'optimisation de forme paramétrique). Dans ce contexte, un ensemble de paramètres variables est également définis mais ceux-ci peuvent agir sur un ensemble de paramètres dimensionnels par exemple. Un choix pertinent de ces paramètres variables peut conduire à la simplification de la construction du problème en assurant une élaboration



(a) Optimisation de forme paramétrique

(b) Optimisation de forme géométrique

(c) Optimisation topologique

FIGURE 1.2 – Différents types d'optimisation de forme [DUYSINX 1996]

plus aisée des solutions admissibles. A l'instar de l'optimisation de forme paramétrique, l'optimisation de forme géométrique peut conduire à des solutions *a priori* admissibles s'avérant impossibles à évaluer par la simulation numérique (en raison, par exemple, de l'impossibilité à mailler la géométrie ou parce que le problème n'est numériquement pas résolvable).

■ **L'optimisation de forme topologique** [BENDSØE 1995 ; BENDSØE et SIGMUND 2003] où l'on cherche la meilleure forme possible au sens des critères retenus. La particularité de ce type d'optimisation est qu'elle autorise et est capable de gérer les modifications de la topologie de la pièce (avec par exemple l'apparition de trous dans la pièce qui initialement étaient absents).

2.1.2 Optimisation paramétrique

L'optimisation paramétrique s'inscrit dans un cadre beaucoup plus vaste que l'optimisation de forme et elle peut être mise en œuvre dans n'importe quel domaine. L'objectif est de trouver la meilleure solution et les paramètres qui lui sont associés au sens d'un critère. Dans le contexte de la conception mécanique, ce type d'optimisation ne se limite pas à l'optimisation de forme et s'étend à tous les cas de figure requérant le choix de paramètres. Par exemple, on peut employer ce type d'optimisation pour chercher à minimiser le prix ou le poids d'une pièce ou d'un ensemble de pièces en considérant comme paramètres les caractéristiques matériaux des pièces.

Les travaux présentés dans ce thèse s'inscrivent dans le cadre de ce type d'optimisation.

2.2 Formulation d'un problème d'optimisation paramétrique

Avant de présenter les différentes stratégies permettant la résolution d'un problème d'optimisation paramétrique, les notions relatives à ce type d'optimisation sont rappelées dans cette partie. Par la suite, quelques éléments classiques permettant de comprendre les aspects importants à prendre en compte lors de la formulation d'un problème d'optimisation seront brièvement présentés. Cette partie n'a pas pour but de fournir un fond mathématique exhaustif mais permet de comprendre les principes clés. Elle a été écrite en s'appuyant entre autres sur [NOCEDAL et S. J. WRIGHT 1999].

2.2.1 Formulation générale et vocabulaire

On considère le problème d'optimisation **sans contrainte** suivant :

Problème 1.1 (Optimisation sans contrainte)
Trouver $(\mathbf{x}^*, y(\mathbf{x}^*)) \in \mathbb{R}^{n_p} \times \mathbb{R}$, solution de : $\min\limits_{\mathbf{x}\in\mathbb{R}^{n_p}} y(\mathbf{x})$

Soit $\mathbf{x}$ le vecteur des n_p paramètres (aussi souvent appelé **jeu de paramètres**) tel que $\mathbf{x} = \{x_1, x_2, \ldots, x_{n_p}\}$ où les x_i sont les paramètres. Dans le contexte de l'optimisation de structures présentée dans ce document, on les désignera également sous le terme **paramètres de conception** (les termes *variables de conception* et *variables de décisions* sont également classiquement utilisés). Lorsque l'on a affaire à plusieurs jeux de paramètres, ceux-ci seront notés $\mathbf{x}^{(k)}$ (avec $k \in \mathbb{N}$).

L'objectif de ce problème consiste à trouver le minimum ou les minima $\mathbf{x}^*$ de la fonction y que l'on nommera **fonction objectif** (également désignée sous le terme **fonction coût**) dans la suite de ce document. La fonction $y : \mathbb{R}^{n_p} \to \mathbb{R}$ est supposée au moins $\mathcal{C}^0$ sur $\mathbb{R}^{n_p}$ [1].

Généralement les paramètres ne varient pas sur $\mathbb{R}$ mais sur un espace borné. Le problème 1.2 d'optimisation sans contrainte ne permet pas de prendre en compte ce type de conditions tel quel. Il est alors nécessaire de considérer un problème d'optimisation **sous contraintes** qui s'écrit dans le cadre général sous la forme suivante :

Problème 1.2 (Optimisation sous contraintes)

Trouver $(\mathbf{x}^*, y(\mathbf{x}^*)) \in \mathbb{R}^{n_p} \times \mathbb{R}$, solution de :

$$\min_{\mathbf{x}\in\mathbb{R}^{n_p}} y(\mathbf{x})$$
$$\text{tout en respectant} \quad \begin{array}{l} \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \end{array}$$

L'objectif est ici de trouver le minimum ou les minima $\mathbf{x}^*$ de la fonction objectif y sous les **contraintes** décrites par les relations $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ et $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$.

Les contraintes peuvent être de deux types :

1. La fonction est continue et différentiable sur $\mathbb{R}^{n_p}$.

- elles peuvent être définies par des égalités telles que celles introduites par la fonction supposée continue $\mathbf{h} : \mathbb{R}^{n_p} \to \mathbb{R}^{n_{ce}}$. L'écriture $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ représente n_{ce} **contraintes d'égalités** que l'on peut également écrire sous la forme : $h_k(\mathbf{x}) = 0$ avec $k \in [\![0, n_{ce}]\!]$;
- elles peuvent être également définies par des inégalités telles que celles introduites par la fonction supposée continue $\mathbf{g} : \mathbb{R}^{n_p} \to \mathbb{R}^{n_{ci}}$. L'écriture $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ représente n_{ci} **contraintes d'inégalités** que l'on peut également écrire sous la forme : $\mathbf{g}_k(\mathbf{x}) \leq 0$ avec $k \in [\![0, n_{ci}]\!]$.

Dans le contexte où l'on souhaite prendre en compte le fait que les paramètres sont bornés, il est possible en employant l'écriture proposée ci-dessus d'écrire n_p groupes de contraintes d'inégalité se formulant, pour chaque paramètre, $x_i - b_i^u \leq 0$ et $b_i^l - x_i \leq 0$ où b_i^u et b_i^l sont respectivement les bornes supérieures et inférieures du paramètres x_i. Au bilan, la prise en compte de ces bornes pour chacun des paramètres de conception consiste à intégrer $2 \times n_p$ contraintes d'inégalités.

D'un point de vue pratique, on considérera que l'on cherche les paramètres $\mathbf{x}$ dans un espace $\mathcal{D}$ dénommé **espace de conception** ($\mathcal{D} \subset \mathbb{R}^{n_p}$) qui définira les bornes relatives à chacun des paramètres. L'inégalité $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ n'intégrera plus ces conditions de bornitude.

Le problème d'optimisation sous contraintes 1.2 s'écrira alors :

Problème 1.3 (Optimisation sous contraintes)

Trouver $(\mathbf{x}^*, y(\mathbf{x}^*)) \in \mathcal{D} \times \mathbb{R}$, solution de :

$$\min_{\mathbf{x} \in \mathcal{D}} y(\mathbf{x})$$
$$\text{tout en respectant} \quad \begin{array}{l} \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \end{array}$$

2.2.2 Solutions du problème d'optimisation et vocabulaire

Classiquement il est possible de définir deux types de minima solutions du problème d'optimisation introduit précédemment : le **minimum global** et le **minimum local** au sens des définitions suivantes.

Soit $X = \{\mathbf{x} \in \mathcal{D} / \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$ l'ensemble des contraintes.

Définition 1.1 (Minimum global) **:** Soit $y : X \to \mathbb{R}$ une fonctionnelle. $\mathbf{x}^* \in \mathbb{R}^{n_p}$ est un point de minimum global si et seulement si,

$$\mathbf{x}^* \in X \text{ et } \forall \mathbf{y} \in X,\ y(\mathbf{x}^*) \leq y(\mathbf{y}) \tag{1.1}$$

On dit alors que $y(\mathbf{x}^*)$ est un **minimum global** de y sur X.

Définition 1.2 (Minimum local) **:** Soit $y : X \to \mathbb{R}$ une fonctionnelle. $\mathbf{x}^* \in \mathbb{R}^{n_p}$ est un point de minimum local si et seulement si,

$$\mathbf{x}^* \in X \text{ et } \exists \boldsymbol{\varepsilon} \in \mathbb{R}^+ \text{ tels que } \forall \mathbf{y} \in [\mathbf{x}^* - \boldsymbol{\varepsilon}, \mathbf{x}^* + \boldsymbol{\varepsilon}] \subset X,\ y(\mathbf{x}^*) \leq y(\mathbf{y}) \tag{1.2}$$

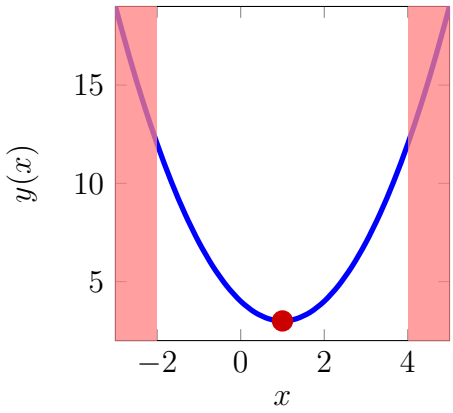
On dit alors que $y(\mathbf{x}^*)$ est un **minimum local** de y sur X.

Définition 1.3 (Minimum local strict) **:** Soit $y : X \to \mathbb{R}$ une fonctionnelle. $\mathbf{x}^* \in \mathbb{R}^{n_p}$ est un point de minimum local strict si et seulement si,

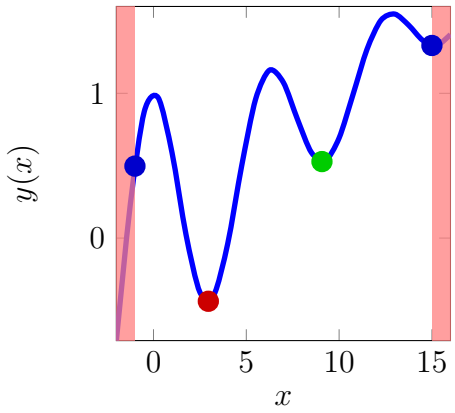
$$\mathbf{x}^* \in X \text{ et } \exists \boldsymbol{\varepsilon} \in \mathbb{R}^+ \text{ tels que } \forall \mathbf{y} \in [\mathbf{x}^* - \boldsymbol{\varepsilon}, \mathbf{x}^* + \boldsymbol{\varepsilon}] \subset X,\ y(\mathbf{x}^*) < y(\mathbf{y}) \qquad (1.3)$$

On dit alors que $y(\mathbf{x}^*)$ est un **minimum local strict** de y sur X.

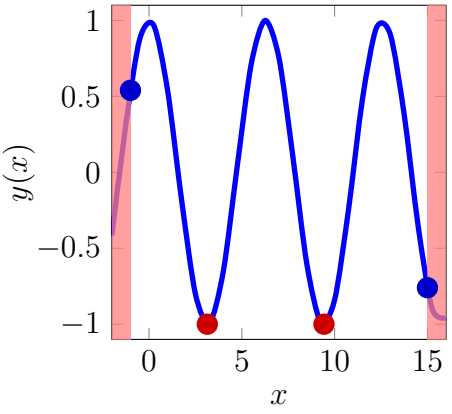
La fonction objectif peut avoir plusieurs minima locaux sur X atteints en des jeux de paramètres différents. Elle peut avoir également plusieurs minima locaux atteints en des points différents mais tels que la valeur de la fonction en ces points soit la même. La figure 1.3 présente différentes fonctions unidimensionnelles ayant différents types de minima sur des espaces bornés.



(a) Fonction $(p-1)^2+3$: 1 minimum global pour $\mathcal{D} = [2, 4]$



(b) Fonction $\exp(-p/10)\cos(p) + 1/10p$: 1 minimum global et 3 minima locaux pour $\mathcal{D} = [-1, 15]$



(c) Fonction $\cos(p)$: 2 minima globaux et 2 minima locaux pour $\mathcal{D} = [-1, 15]$

FIGURE 1.3 – Quelques exemples de minima globaux (●) et locaux (●et●)

Sur ces exemples, on observe des minima globaux (●) et locaux (●et●). Il est important

de noter que la restriction de l'espace de recherche à un espace borné peut conduire à l'apparition de minima sur les bords de cet espace (cf. figures 1.3a et 1.3b minima locaux ●). Par ailleurs, la recherche de minima globaux sur les fonctions représentées sur les figures 1.3b et 1.3c lors d'un problème d'optimisation **sans** contrainte (on cherche les minima sur $\mathbb{R}$) ne peuvent clairement aboutir car ces problèmes comportent pour le premier cas (figure 1.3b) une infinité de minima locaux et pas de minimum global et pour le second (figure 1.3c) une infinité de minima globaux sur $\mathbb{R}$.

Par ailleurs, la présence de fonctions contraintes peut avoir un rôle extrêmement important sur l'obtention du minimum. Par exemple, la figure 1.4 présente une fonction de deux variables [2] ainsi que les minima du problème sans et avec une contrainte d'inégalité ($12x_2 + x_1^2 + 6x_1 + 9 > 0$) : dans le cas présenté, la contrainte d'inégalité entraîne un changement de minimum global (cf. minima ● sur les figures 1.4b et 1.4c).

2.2.3 Quelques éléments relatifs à l'unicité et à l'existence du minimum

Dans le cadre de l'optimisation considérée ici, il est possible de connaître *a priori* l'existence et l'unicité de la solution à un problème d'optimisation sans et sous contraintes (à condition bien entendu de connaître certaines propriétés de la fonction objectif et des fonctions contraintes).

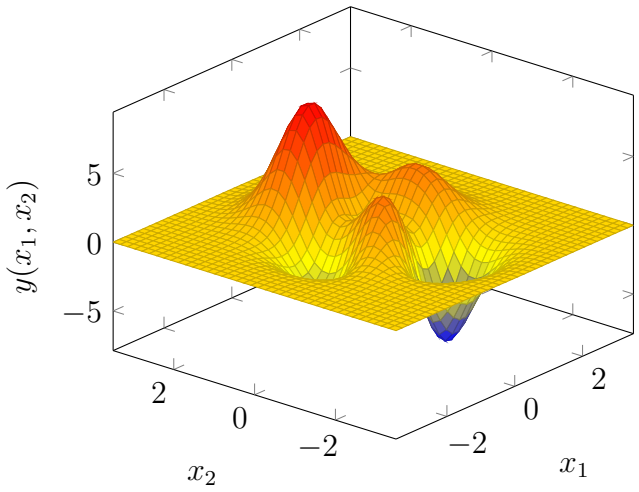
Ces résultats d'existence sont regroupés sous forme de **conditions d'optimalité** dites du premier et du second ordre (portant respectivement sur le gradient et le hessien de la fonction objectif). Il est possible de les écrire pour chaque type de problèmes (optimisation sans contrainte, optimisation avec contraintes d'égalités, optimisation avec contraintes d'inégalités et optimisation avec contraintes d'égalités et d'inégalités). Dans le cadre le plus général, c'est-à-dire dans le cadre d'un problème d'optimisation avec contraintes d'égalités et d'inégalités, on emploie classiquement les conditions nécessaires d'optimalité connues sous le nom de **conditions de Karush-Kuhn-Tucker** [KARUSH 1939 ; KUHN et TUCKER 1951]. Ces conditions ne seront pas explicitées ici mais peuvent être trouvées dans [NOCEDAL et S. J. WRIGHT 1999] par exemple.

Certains des algorithmes d'optimisation ont été conçus et/ou leurs convergences prouvées à partir des conditions d'optimalité. Parmi les algorithmes dont la convergence a été prouvée (sous certaines conditions), on retrouve les algorithmes de descente de gradient avec pas de Wolfe, avec pas fixe et avec pas optimal, les algorithmes de gradients conjugués (par exemple Pollack-Rivière), les algorithmes de Newton et de quasi-Newton (par exemple BFGS)...

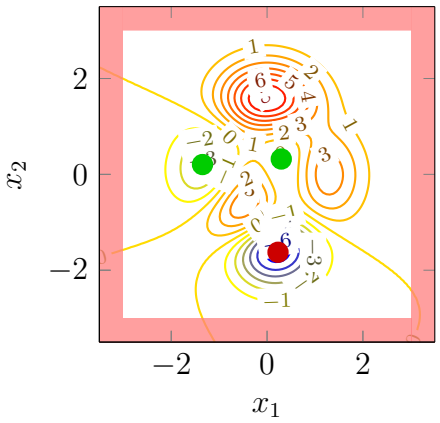
2.2.4 Problématiques de l'optimisation

Un problème d'optimisation tel que celui introduit par l'équation 1.2 ne peut être résolu de manière analytique qu'à condition de connaître la dépendance de la fonction objectif et des fonctions contraintes (s'il y en a) aux paramètres de manière explicite et que celle-ci

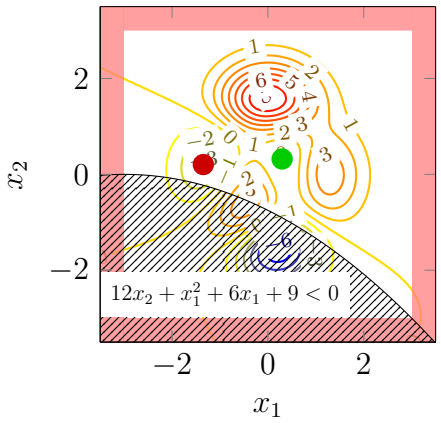
2. $y(x_1, x_2) = 3(1-x_1)^2 \exp[-x_1^2 - (x_2+1)^2] - 10(x_1/5 - x_1^3 - x_2^5) \exp[-x_1^2 - x_2^2] - 1/3 \exp[-(x_1+1)^2 - x_2^2]$



(a) Fonction Peaks



(b) Fonction et minima sur $\mathcal{D}^2 = [-3, 3]^2$



(c) Fonction et minima avec contrainte d'inégalité

FIGURE 1.4 – Exemple de fonction à 2 variables et minima associés avec et sans contrainte d'inégalité (globaux ● et locaux ●)

ne soit pas trop complexe. Dans ce cas, l'emploi de méthodes analytiques de recherche des extremums de la fonction est envisageable.

Malheureusement, dans le contexte de l'optimisation mécanique (ce qui vaut aussi dans d'autres domaines), la fonction objectif à minimiser ne dépend pas de manière simple des paramètres de conception considérés. Par exemple, si l'on considère une dimension de la structure comme étant un paramètre de conception et que l'on cherche à minimiser la contrainte de cisaillement dans celle-ci, la solution du problème mécanique sera dépendante de cette dimension sans pour autant permettre l'écriture d'une fonction analytique donnant la contrainte de cisaillement en fonction de la dimension. Dans cet exemple, la modification de la dimension peut avoir un impact sur le maillage de la structure (modification de la forme des éléments, ajout/retrait d'éléments) nécessaire à la résolution numérique par éléments finis (par exemple). Cette modification du maillage aura pour conséquence de modifier les grandeurs intervenant dans la résolution numérique du problème mécanique (telles que la matrice de rigidité, celle de masse, le vecteur des efforts nodaux...).

Ainsi dans le contexte de l'optimisation d'assemblages, l'optimisation numérique s'avère indispensable et celle-ci devra être robuste vis-à-vis des différents problèmes d'optimisation qui pourront être rencontrés. En particulier, des problèmes d'optimisation avec et sans contraintes faisant intervenir des fonctions objectifs et contrainte présentant des non-linéarités plus ou moins marquées pourront être traités.

2.3 Algorithmes d'optimisation

Dans cette partie les algorithmes d'optimisation classiquement employés seront rappelés brièvement. On ne s'intéressera qu'aux méthodes permettant de résoudre un problème d'optimisation déterministe.

2.3.1 Optimisation locale ou globale ?

Comme il a été vu précédemment, la recherche de minima d'une fonction peut faire apparaître deux types de minima : des minima globaux et locaux. On caractérisera de multimodal un problème comportant plusieurs minima.

Dans le cadre des travaux décrits ici, on cherche à trouver le minimum global de la fonction objectif sans pour autant être pénalisé par la présence de minima locaux. En effet, parmi les méthodes d'optimisation numérique, on peut distinguer deux catégories : les méthodes capables de trouver le minimum global d'une fonction tout en étant capable de s'affranchir des minima locaux et celles dédiées à la recherche locale de minima autour d'un point initial. Les algorithmes de la deuxième catégorie, du fait de leur comportement, ne parviennent généralement pas à trouver le minimum global d'une fonction si le point initial n'est pas judicieusement choisi.

2.3.2 Algorithmes d'optimisation locale

La recherche de minima locaux nécessite l'emploi d'algorithmes n'assurant pas ou très peu de recherche exploratoire mais exploitant au maximum les informations relatives à

l'évolution de la fonction objectif. Ils sont initialisés en un jeu de paramètres donnés et vont chercher à évoluer autour de ce dernier en utilisant des règles d'évolution lui permettant de se diriger vers le minimum le plus proche du point initial.

Quelques algorithmes classiques d'optimisation locale seront décrits brièvement dans la suite. Dans un premier temps, seuls les algorithmes d'optimisation locale **sans** contrainte seront introduits. Des stratégies et des algorithmes dédiés pour résoudre des problèmes d'optimisation locale **avec** contraintes seront ensuite rappelés.

Parmi les méthodes d'optimisation locale, on peut distinguer :

- les **méthodes de recherche directe** (*direct search* [M. H. WRIGHT 1996 ; R. M. LEWIS, TORCZON et TROSSET 2000 ; KOLDA, R. M. LEWIS et TORCZON 2003]) qui utilisent uniquement les réponses de la fonction pour localiser le minimum et n'exploitent pas le gradient de la fonction objectif (y compris un gradient calculé par différences finies). Les avantages de ces méthodes reposent sur leur simplicité, leur robustesse et leur efficacité, y compris sur des fonctions bruitées (voir par exemple [MCKINNON 1998]) dans un contexte d'optimisation à variables continues sans contrainte. Leur principal inconvénient réside dans le fait qu'il n'existe pas de propriétés théoriques de convergence pour la majorité d'entre elles (à deux exceptions près, voir [TORCZON 1991 ; TORCZON 1997]). Les méthodes de recherche directe ont été introduites dans les années 1950-1960 et peuvent être classées en trois grandes catégories :

  - les méthodes de recherche par motifs généralisés (*evolutionary operation* [BOX 1957], *pattern search method* [HOOKE et JEEVES 1961], *multidirectional search algorithm* [TORCZON 1989 ; J. E. DENNIS J. et TORCZON 1991 ; TORCZON 1991], *generalized pattern search method* [TORCZON 1997]) utilisant un ensemble de directions prédéfinies dans l'espace des paramètres afin de faire évoluer l'algorithme vers la zone où se trouve le minimum.
  - les méthodes des directions conjuguées [ROSENBROCK 1960 ; M. J. POWELL 1964 ; M. J. POWELL 1998 ; FRIMANNSLUND et STEIHAUG 2007] où les directions dans l'espace des paramètres sont définies au fur et à mesure des itérations. Elles cherchent à accélérer la convergence de l'algorithme en tenant compte des informations sur la courbure de la fonction objectif.
  - les méthodes basées sur un simplexe (exemple : algorithme de Nelder-Mead [NELDER et MEAD 1965] et ses variantes [WALTERS et al. 1991]) qui utilisent une figure géométrique (simplexe) pour localiser le minimum. L'algorithme fait subir au simplexe un ensemble d'évolutions (expansion, contraction, rétrécissement) afin de l'adapter au comportement local de la fonction objectif.

- les **méthodes de descente** qui consistent, à partir d'un jeu de paramètres initial, à trouver itérativement des jeux de paramètres améliorant la valeur de la fonction [CAUCHY 1847]. Ayant accès à la valeur de la fonction et de ses dérivées pour un jeu de paramètres donnés, le principe consiste à chercher à l'itération $i + 1$ le jeu de

paramètres $\mathbf{x}^{(i+1)}$ tel que $y\big(\mathbf{x}^{(i+1)}\big) < y\big(\mathbf{x}^{(i)}\big)$. L'algorithme tente alors de converger itérativement vers le minimum local selon une direction (on parle de *direction de descente*) et une amplitude (le *pas*) calculées par l'algorithme. Ainsi, à l'itération $i+1$ de l'algorithme, le jeu de paramètres est calculé de la manière suivante : $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + h_i\mathbf{u}_i$ où h_i et $\mathbf{u}_i$ sont le pas et la direction de descente. L'algorithme détermine h_i et $\mathbf{u}_i$ tels que $\boldsymbol{\nabla} y(\mathbf{x}^{(i)})^T\mathbf{u}_i < 0$ et $y\big(\mathbf{x}^{(i)} + h_i\mathbf{u}_i\big) < y\big(\mathbf{x}^{(i)}\big)$. Parmi ces méthodes de descente, on distingue différentes méthodes « classiques » employant le gradient de la fonction objectif.

- ▷ l'algorithme de gradient à pas constant où la direction de descente est choisie opposée à la direction du gradient soit $\mathbf{u}_i = -\boldsymbol{\nabla} y(\mathbf{x}^{(i)})$. Le pas de la descente est choisi par l'algorithme arbitrairement et de telle manière qu'il soit constant. Un choix judicieux de ce paramètre est cependant indispensable pour assurer la convergence de l'algorithme. En effet, prendre un pas constant trop important peut rendre la convergence impossible. À l'inverse, choisir un pas trop faible peut entraîner une convergence très lente de l'algorithme.
- ▷ l'algorithme de gradient à pas optimal aussi désigné sous le terme d'algorithme de plus grande pente (*steepest descent*) [CAUCHY 1847] où le pas de la descente est choisi de manière à obtenir la descente la plus performante. Le pas est cherché tel que $h_i = \arg\min_h y(\mathbf{x}^{(i)} - h\mathbf{u}_i)$ avec $\mathbf{u}_i = -\boldsymbol{\nabla} y(\mathbf{x}^{(i)})$. On parle alors de pas optimal. La recherche du pas optimal (on désigne cette étape par *recherche linéaire* ; cf. remarque 1.1) à chaque itération peut s'avérer extrêmement coûteuse et il n'est pas possible en pratique d'en déterminer une valeur exacte mais seulement une valeur approchée en utilisant par exemple les conditions de Wolfe [WOLFE 1969 ; WOLFE 1971].
- ▷ l'algorithme de gradients conjugués [HESTENES et STIEFEL 1952] initialement introduit pour résoudre des systèmes linéaires et qui en assure la résolution en un nombre d'itérations égal au maximum à la dimension du problème (ce nombre peut être réduit en utilisant un préconditionneur) [SHEWCHUK 1994]. Dans le cadre de l'optimisation [FLETCHER et REEVES 1964], l'idée consiste à considérer à l'itération i comme direction de descente $\mathbf{u}_i = -\boldsymbol{\nabla} y(\mathbf{x}^{(i)}) - \beta_i\mathbf{u}_{i-1}$ (si $i > 0$ et $\mathbf{u}_0 = -\boldsymbol{\nabla} y(\mathbf{x}^{(0)})$) avec $\beta_i = \frac{\|\boldsymbol{\nabla} y(\mathbf{x}^{(i)})\|^2}{\|\boldsymbol{\nabla} y(\mathbf{x}^{(i-1)})\|^2}$ (algorithme de Pollack-Rivière, voir par exemple [GILBERT et NOCEDAL 1992]). On détermine ensuite le pas optimal dans cette direction.

- ▶ les **méthodes de Newton** qui cherchent les solutions de l'équation $\boldsymbol{\nabla} y(\mathbf{x}) = 0$; c'est-à-dire les points critiques de la fonction objectif. Pour cela, elles emploient le hessien $\boldsymbol{\nabla}^2 y(\mathbf{x})$ de la fonction objectif (en supposant que la fonction de classe $\mathcal{C}^2$ et que la matrice hessienne $\boldsymbol{\nabla}^2 y(\mathbf{x})$ soit inversible) :
  - ▷ l'algorithme de Newton basé sur la construction d'une approximation quadratique de la fonction objectif au voisinage du point considéré à l'itération i : $y\big(\mathbf{x}^{(i)} + \mathbf{v}\big) \simeq y\big(\mathbf{x}^{(i)}\big) + \boldsymbol{\nabla} y(\mathbf{x}^{(i)})\mathbf{v} + \frac{1}{2}\mathbf{v}^T\boldsymbol{\nabla}^2 y(\mathbf{x}^{(i)})\mathbf{v}$ dont le gradient

s'annule pour $\mathbf{v} = -[\boldsymbol{\nabla}^2 y(\mathbf{x}^{(i)})]^{-1}\boldsymbol{\nabla} y(\mathbf{x}^{(i)})$. L'algorithme de Newton cherche itérativement la solution telle que $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - [\boldsymbol{\nabla}^2 y(\mathbf{x}^{(i)})]^{-1}\boldsymbol{\nabla} y(\mathbf{x}^{(i)})$ (le terme $d_i = [\boldsymbol{\nabla}^2 y(\mathbf{x}^{(i)})]^{-1}\boldsymbol{\nabla} y(\mathbf{x}^{(i)})$ est appelé direction de Newton). Le principal inconvénient de cet algorithme est qu'il requiert à chaque itération le calcul et l'inversion coûteux de la matrice hessienne. Cette inversion peut par ailleurs être réellement problématique en cas de mauvais conditionnement de la matrice hessienne.

▷ l'algorithme de quasi-Newton qui approche le calcul de l'inverse de la matrice hessienne par les méthodes DFP [DAVIDON 1959 ; FLETCHER et M. J. D. POWELL 1963] ou BFGS [BROYDEN 1970 ; FLETCHER 1970 ; GOLDFARB 1970 ; SHANNO 1970] telle que, respectivement, en posant $\mathbf{h}_i \approx [\boldsymbol{\nabla}^2 y(\mathbf{x}^{(i)})]^{-1}$:

$$\mathbf{h}_{i+1}^{DFP} = \mathbf{h}_i^{DFP} + \frac{\boldsymbol{\delta}_i \boldsymbol{\delta}_i^T}{\boldsymbol{\delta}_i^T \boldsymbol{\gamma}_i} - \frac{\mathbf{h}_i^{DFP} \boldsymbol{\gamma}_i \boldsymbol{\gamma}_i^T \mathbf{h}_i^{DFP}}{\boldsymbol{\gamma}_i^T \mathbf{h}_i^{DFP} \boldsymbol{\gamma}_i}$$

$$\mathbf{h}_{i+1}^{BFGS} = \mathbf{h}_i^{BFGS} + \left(1 + \frac{\boldsymbol{\gamma}_i^T \mathbf{h}_i^{BFGS} \boldsymbol{\gamma}_i}{\boldsymbol{\delta}_i^T \boldsymbol{\gamma}_i}\right) \frac{\boldsymbol{\delta}_i \boldsymbol{\delta}_i^T}{\boldsymbol{\delta}_i^T \boldsymbol{\gamma}_i} - \frac{\boldsymbol{\delta}_i \boldsymbol{\gamma}_i^T \mathbf{h}_i^{BFGS} + \mathbf{h}_i^{BFGS} \boldsymbol{\gamma}_i \boldsymbol{\delta}_i^T}{\boldsymbol{\delta}_i^T \boldsymbol{\gamma}_i}$$

avec $\boldsymbol{\gamma}_i = \boldsymbol{\nabla} y(\mathbf{x}^{(i+1)}) - \boldsymbol{\nabla} y(\mathbf{x}^{(i)})$ et $\boldsymbol{\delta}_i = \mathbf{x}^{(i+1)} - \mathbf{x}^{(i)}$. Le principal avantage de ces méthodes d'approximation est qu'elles ne nécessitent que le calcul du gradient de la fonction objectif et permettent de profiter des très bonnes performances de l'algorithme de Newton.

Classiquement, dans un contexte d'optimisation locale sans contrainte, les méthodes de descente basées sur l'emploi du gradient et les méthodes de Newton sont les plus utilisées. Ces méthodes étant pour la plupart des méthodes paramétriques, leur mise en œuvre nécessite une attention particulière afin de s'assurer qu'elles convergent correctement. Par ailleurs, leur emploi dans un contexte d'optimisation locale sans contrainte sur une fonction non nécessairement convexe peut engendrer l'obtention d'un minimum local qui ne correspond pas à celui cherché. Le choix d'un point d'initialisation devra donc être fait de manière judicieuse.

Remarque 1.1 : Dans le cadre de l'emploi de méthodes de descente, la méthode de détermination du pas de la descente se fait par l'emploi d'une méthode de recherche linéaire (*line search*). Il existe une approche plus évoluée consistant à employer une méthode d'approximation quadratique locale permettant une détermination couplée du pas et de la direction de descente basée sur l'emploi de cette approximation. Cette méthode est connue sous le nom de **méthode de région de confiance** [SORENSEN 1982 ; BYRD, SCHNABEL et SHULTZ 1987 ; M. POWELL et YUAN 1990]. Elle fut par la suite adaptée à de nombreuses autres méthodes d'optimisation.

2.3.3 Algorithmes d'optimisation locale sous contraintes

Dans le cadre de l'optimisation locale, il est nécessaire d'être capable de prendre en compte les contraintes. De deux types, ces contraintes jouent alors un rôle prépondérant sur la procédure d'optimisation. Elles nécessitent l'usage de stratégies dédiées et ont un impact sur le problème en lui même. En particulier, à chaque itération de l'algorithme, il doit être vérifié que le jeu de paramètres respecte les contraintes (qu'il ne les « viole » pas). Les très nombreuses méthodes d'optimisation sous contraintes peuvent être catégoriser en fonction des spécificités des fonctions objectifs ($y(\mathbf{x})$) et fonctions contraintes (contraintes d'égalités $\mathbf{h}(\mathbf{x})$ et d'inégalités $\mathbf{g}(\mathbf{x})$) [NOCEDAL et S. J. WRIGHT 1999] :

- la **programmation linéaire** (*Linear Programming*) permettant de résoudre des problèmes où la fonction objectif et toutes les fonctions contraintes sont linéaires.
- la **programmation quadratique** (*Quadratic Programming*) où les fonctions contraintes sont linéaires et la fonction objectif est quadratique.
- la **programmation non-linéaire** (*Nonlinear Programming*) lorsqu'au moins l'une des fonctions contraintes est non-linéaire.
- l'**optimisation à contraintes linéaires** (*Linear constrained optimization*) qui, comme son nom l'indique, propose de traiter spécifiquement les problèmes d'optimisation ayant des contraintes linéaires.
- l'**optimisation avec contraintes de borne** permettant de résoudre un problème d'optimisation où les paramètres sont bornés (définissant l'espace de conception $\mathcal{D}$).
- la **programmation convexe** où la fonction objectif est convexe, les contraintes d'égalités sont linéaires et les contraintes d'inégalités sont concaves.

Ce classement est plutôt à considérer comme une grille de lecture des différentes et nombreuses désignations employées dans le cadre de l'optimisation. Il n'est pas exclusif ni exhaustif. Parmi les algorithmes spécifiquement conçus pour assurer la résolution de problèmes d'optimisation sous contraintes, on trouve entre autres :

- les **méthodes de pénalisation** qui consistent à reformuler la fonction objectif en y intégrant les contraintes de manière à ce que la nouvelle fonction à minimiser s'écrive : $y_p(\mathbf{x}) = y(\mathbf{x}) + \frac{1}{\epsilon} p(\mathbf{x})$ où $p(\mathbf{x})$ est une fonction de pénalité prenant des valeurs importantes lorsque les contraintes ne sont pas vérifiées. L'avantage de ces méthodes réside dans le fait qu'elles permettent de transformer le problème d'optimisation sous contraintes en un problème d'optimisation sans contrainte portant sur la nouvelle fonction $y_p(\mathbf{x})$. Dans la pratique, les méthodes de pénalisation sont particulièrement complexes à mettre en œuvre car elles peuvent conduire à des problèmes mal conditionnés. Par ailleurs, le choix de la valeur de ϵ est particulièrement difficile et joue un rôle prépondérant dans la résolution du problème d'optimisation.

- les **méthodes lagrangiennes** qui consistent à former une fonctionnelle nommée Lagrangien associée au problème d'optimisation en faisant intervenir la fonction objectif et les fonctions contraintes. Le nouveau problème à résoudre fait alors intervenir non seulement les paramètres de conception mais également des multiplicateurs de Lagrange λ_i. Le Lagrangien s'écrit alors : $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = y(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{x})$ où $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \dots & \lambda_{n_c} \end{bmatrix}^T$. Le problème d'optimisation initial visant à trouver le minimum de la fonction objectif sous les contraintes $c_i(\mathbf{x}) \leq 0$ avec $i \in [\![1, n_c]\!]$ devient alors un problème d'optimisation sans contrainte pouvant être résolu à l'aide d'un algorithme d'optimisation sans contrainte comme l'un de ceux introduits précédemment (cf. partie 2.3.2 page 19). Par ailleurs, ce type de problèmes revient à un problème de recherche de point selle que l'on peut résoudre par la méthode d'Uzawa [ARROW, HURWICZ et UZAWA 1958].

- Dans le cas où l'on souhaiterait utiliser une méthode de descente, il est nécessaire de trouver à chaque itération de l'algorithme un nouveau jeu de paramètres dit « admissible » c'est-à-dire qu'il tend à minimiser la fonction objectif tout en respectant la ou les contrainte(s). Il est alors possible d'employer :

  - l'algorithme du gradient à pas fixe avec projection ou **algorithme du gradient projeté** [GOLDSTEIN 1964 ; LEVITIN et POLYAK 1966] qui fait appel à un projecteur permettant d'éviter que l'algorithme ne « viole » une ou plusieurs contraintes. Le nouveau jeu de paramètres à chaque itération de l'algorithme est alors déterminé de la manière suivante : $\mathbf{x}^{(i+1)} = \mathcal{P}(\mathbf{x}^{(i)} - h_i \boldsymbol{\nabla} y(\mathbf{x}^{(i)}))$ où $\mathcal{P}$ assure la projection de $\mathbb{R}^{n_p}$ sur l'espace $\mathcal{D}_c$ des vecteurs $\mathbf{x}$ admissibles tel que $\mathcal{D}_c = \mathcal{D} \cap \{\mathbf{x} / \forall i \in [\![1, n_c]\!],\ c_i(\mathbf{x}) \leq 0\}$. Un des principaux inconvénients de cette méthode réside dans le fait que le projecteur peut-être particulièrement complexe à déterminer. C'est pourquoi cette méthode est peu employée en pratique.
  - la **méthode des directions admissibles** [ZOUTENDIJK 1960 ; ELWAKEIL et ARORA 1995 ; X. CHEN 1999 ; X. CHEN et KOSTREVA 1999] consiste à déterminer le nouveau jeu de paramètres à chaque itération de l'algorithme tel que, à l'itération $i+1$, $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + h_i \mathbf{u}_i$. La direction de descente $\mathbf{u}_i$ est déterminée de manière à minimiser $\boldsymbol{\nabla} y(\mathbf{x}^{(i)})^T \mathbf{u}_i$ sous les contraintes $\forall k \in [\![1, n_c]\!]$, $\boldsymbol{\nabla} c_k(\mathbf{x}^{(i)})^T \mathbf{u}_i \leq 0$. Une fois la direction obtenue (on parle alors de direction admissible et non plus de direction de descente), il reste à déterminer le pas dans la direction admissible en employant une méthode d'optimisation unidimensionnelle. Lors de la recherche de la direction admissible, on ajoute une contrainte de bornitude telle que $\|\mathbf{u}_i\|_1 = 1$ ou $\|\mathbf{u}_i\|_2 = 1$ [3] afin de se prémunir des solutions infinies.

- Les algorithmes de Newton présentés dans la partie 2.3.2 page 21 sont basés sur la linéarisation d'équations caractérisant les solutions que l'on cherche. Ces méthodes

3. $\| \bullet \|_1$ et $\| \bullet \|_2$ désignent les normes 1 et 2.

peuvent être employées afin de résoudre un problème d'optimisation sous contraintes. Cette approche repose sur l'écriture d'un Lagrangien $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = y(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{x})$. L'écriture d'une approximation quadratique du Lagrangien autour du jeu de paramètres considéré à l'itération i conduit à la construction du problème quadratique 1.4 permettant l'obtention de la direction de descente $\mathbf{u}_i$:

$$\begin{aligned} &\min_{\mathbf{u}_i \in \mathbb{R}^{n_p}} \boldsymbol{\nabla} y(\mathbf{x}^{(i)})^T \mathbf{u}_i + \frac{1}{2} \mathbf{u}_i^T \mathbf{h}_i \mathbf{u}_i \\ &\text{tel que } c_k\big(\mathbf{x}^{(i)}\big) + \boldsymbol{\nabla} c_k(\mathbf{x}^{(i)})^T \mathbf{u}_i = 0 \text{ avec } k \in [\![1, n_c]\!] \end{aligned} \tag{1.4}$$

Cette technique connue sous le nom de *Sequential Quadratic Programming* [GARCIA PALOMARES et MANGASARIAN 1976 ; S.-P. HAN 1977 ; BOGGS et TOLLE 1995] présente l'avantage de remplacer le problème initial pouvant être fortement non linéaire en une suite de problèmes quadratiques sous contraintes linéaires beaucoup plus simples à résoudre. La détermination du hessien $\mathbf{h}_i$ peut bien entendu être réalisée à l'aide de techniques similaires à celles employées par les méthodes quasi-newtoniennes, telle que la méthode BFGS.

- les **méthodes de points intérieurs** introduites dans les années 1950 reposent sur l'emploi d'une fonction nommée *fonction barrière logarithmique* [FRISCH 1955 ; FIACCO et MCCORMICK 1968]. Des difficultés de mise en œuvre (en particulier le mauvais conditionnement de la matrice hessienne de la fonction barrière) et les performances de la méthode du simplexe ont conduit à un abandon de cette approche. C'est en 1984 avec les travaux de Karmarkar [KARMARKAR 1984] que les méthodes de points intérieurs réapparaissent pour traiter des problèmes d'optimisation linéaire.

  Dans un contexte d'optimisation sous contraintes d'inégalités, la résolution par la méthode de points intérieurs consiste à formuler un problème de minimisation d'une fonction barrière $\varphi_\mu(\mathbf{x}) = y(\mathbf{x}) - \mu \sum_{i=1}^{n_{ci}} \log(-\mathbf{g}(\mathbf{x}))$. Le principe de résolution consiste alors à minimiser la fonction barrière pour une valeur de μ fixée. On calcule ensuite une nouvelle valeur de ce paramètre $\mu > 0$ inférieur au précédent. Les itérations font alors succéssivement décroître μ qui tend alors vers 0. Les méthodes d'optimisation par points intérieurs font classiquement appel à des méthodes quasi-Newton. Par la suite, de nombreuses extensions de cette méthode permirent le traitement de problèmes d'optimisation comprenant des contraintes d'égalités linéaires et enfin des contraintes d'inégalités et d'égalités non-linéaires [FORSGREN, GILL et M. H. WRIGHT 2002]. On pourra se référer à [M. H. WRIGHT 1992 ; S. J. WRIGHT 1997] pour des détails complémentaires sur les principes intervenant dans ce type d'approches.

Le nombre de méthodes d'optimisation locale est extrêmement important. Celles-ci ont fait l'objet de très nombreux développements et implémentations dans des logiciels libres ou commerciaux. Il n'existe quasiment pas de travaux de comparaisons globales permettant de faire un choix définitif entre des méthodes d'optimisation directes ou non pour chaque type de problèmes. On trouve néanmoins les travaux [RIOS et SAHINIDIS

2013] qui proposent une comparaison de logiciels commerciaux et libres basée sur de nombreux exemples pour le choix d'une méthode d'optimisation directe pour résoudre un problème d'optimisation avec contraintes de borne. Le bilan est assez mitigé dans le sens où la plupart des algorithmes testés fournissent des résultats similaires.

2.3.4 Algorithmes d'optimisation globale

Les méthodes d'optimisation locales introduites précédemment permettent de trouver le point minimum d'un problème d'optimisation avec ou sans contraintes dans le voisinage d'un point initialement choisi. Le défaut principal de ce type de méthodes est qu'elles n'assurent pas d'obtenir le minimum global dans tous les cas. En effet, dans le cas d'une fonction multimodale (ayant plusieurs minima), le minimum atteint sera lié au point initial choisi pour initialiser l'algorithme. Dans le cadre de l'optimisation d'assemblage proposé dans ces travaux, on cherche à trouver le minimum global associé au problème d'optimisation considéré. Pour cela, il est nécessaire d'employer un algorithme d'optimisation globale.

On propose, dans cette partie, de rappeler les principes des différents algorithmes de recherche globale employés couramment. Afin d'expliciter les différents principes régissant les algorithmes d'optimisation globale, il est possible de les classer en distinguant

- les **méthodes énumératives** qui cherchent à détecter les zones de l'espace de recherche qui ne contiennent pas de minimum. Pour cela, elles se basent sur une discrétisation de l'espace de recherche. Le principal inconvénient de ce type de méthodes est qu'elles nécessitent un nombre important d'évaluations pour parvenir à localiser le minimum global (ce nombre est d'autant plus grand que la dimension du problème est importante et que le pas de la discrétisation est fin). Parmi les méthodes énumératives, on trouve
  - la méthode *branch and bound* [LAND et DOIG 1960], qui est une méthode récursive de recherche de minima basée sur le partitionnement de l'espace de recherche en plusieurs zones pour y détecter la présence du minimum. Elle s'applique dans le cadre de l'optimisation discrète. La méthode cherche alors à déterminer des bornes d'encadrement du minimum.
  - la méthode DIRECT (DI*viding* RECT*angles* [D. R. JONES, PERTTUNEN et STUCKMAN 1993]) construisant itérativement un pavage de $\mathcal{D}$. Cet algorithme s'applique aux fonctions lipschitziennes [4]. Les rectangles du pavage sont progressivement divisés en considérant tous les compromis entre la valeur de la fonction objectif au centre de chaque pavé et la taille du pavé.
- les **méthodes de recherche aléatoires** [ANDERSON 1953 ; RASTRIGIN 1963 ; SOLIS et WETS 1981 ; BOENDER et al. 1982] qui consistent à tirer des jeux de paramètres aléatoires et à évaluer la fonction objectif en ces points. À la suite à chaque

4. Soit $E \subset \mathbb{R}$, et $f : E \to \mathbb{R}$, on dit que f est k-lipschitzienne si $\forall (x, y) \in E^2, |f(x) - f(y)| \leq k|x - y|$

évaluation, on compare la valeur obtenue avec la meilleure valeur obtenue précédemment. Si une meilleure valeur est trouvée alors on l'enregistre. Le processus continue ensuite. Ces méthodes peuvent être couplées avec des méthodes de recherche locale où les points tirés au hasard sont utilisés comme points initiaux de l'algorithme d'optimisation locale. Il faut noter que ces méthodes peuvent converger plusieurs fois vers les mêmes minima locaux.

- les **méthodes de regroupement** (*clustering*) [R. W. BECKER et LAGO 1970 ; TORN 1977 ; RINNOOY KAN et TIMMER 1987] qui, suite à un échantillonnage de l'espace de recherche et à l'évaluation de la fonction objectif aux points échantillonnés, regroupent les points en régions prometteuses. Dans chaque zone, une recherche locale est ensuite réalisée. Les recherches locales convergeant vers une zone déjà explorée sont abandonnées. Dans le cas d'une fonction possédant de nombreux minima, ces méthodes nécessitent un nombre important d'échantillons et donc d'évaluations de la fonction objectif pour converger.

- les **méthodes de descente généralisées** basées sur le principe des méthodes de descente brièvement décrites précédemment. Ces stratégies visent à éviter de converger vers un minimum local lorsque celui-ci a déjà été trouvé et à poursuivre la recherche dans une autre zone potentielle où se situerait un « meilleur » minimum. Il existe deux principales stratégies permettant d'éviter de converger vers un minimum connu :

  - il est possible de pénaliser la fonction objectif au fur et à mesure du déroulement du processus d'itération afin de tenir compte des minima déjà atteints : dans le cas où n_m minima auraient déjà été atteints (aux points $\mathbf{x}^{(i)*}$ avec $i \in [\![0, n_m]\!]$), la fonction objectif pénalisée s'écrirait $y_p(\mathbf{x}) = y(\mathbf{x}) + k(\mathbf{x}, \mathbf{x}^{(1)*}, \mathbf{x}^{(2)*}, \ldots, \mathbf{x}^{(n_m)*})$ avec $k(\mathbf{x}, \mathbf{x}^{(1)*}, \mathbf{x}^{(2)*}, \ldots, \mathbf{x}^{(n_m)*}) = \sum_{i=1}^{n_m} K_i / |\mathbf{x} - \mathbf{x}^{(i)*}|$. Cette pénalisation permet d'obtenir $\lim_{\mathbf{x} \to \mathbf{x}^{(i)*}} y_p(\mathbf{x}) = +\infty$.
  - une autre stratégie consiste à chercher un nouveau point d'initialisation [LEVY et MONTALVO 1985] de l'algorithme de descente : suite à la convergence de l'algorithme vers un minimum local $\mathbf{x}^{(i)*}$ à partir d'un point initial $\mathbf{x}_0^{(i)}$, une nouvelle phase entre en jeu. Celle-ci cherche un nouveau point d'initialisation de l'algorithme de descente. Pour cela, on emploie une fonction T permettant l'obtention du nouveau point $\mathbf{x}_0^{(i+1)}$ qui doit vérifier $y\left(\mathbf{x}_0^{(i+1)}\right) < y\left(\mathbf{x}^{(i)*}\right)$. Ce nouveau point permettra alors d'obtenir un « meilleur » minimum. La fonction T, également appelée fonction tunnel, peut par exemple s'écrire : $T(\mathbf{x}) = \left(y(\mathbf{x}) - \min_i y\left(\mathbf{x}^{(i)*}\right)\right) / \prod_{i=1}^{n_m} \|\mathbf{x} - \mathbf{x}^{(i)*}\|^{2\alpha_i}$ où n_m est le nombre de minima déjà atteints et α_i les paramètres de répulsion (contrôlant l'éloignement des minima déjà atteints pour la recherche du nouveau jeu de paramètres).

- les **méthodes évolutionnaires** sont des méthodes stochastiques qui peuvent être présentées par analogie avec la théorie de la sélection naturelle de Darwin : les individus les plus performants d'une population ont une plus grande probabilité de survivre et

d'engendrer des descendants encore plus résistants. Le mimétisme de ce constat peut être obtenu numériquement en considérant les individus de la population comme des jeux de paramètres $\mathbf{x}^{(k)}$ et leur performance est mesurée par la fonction objectif $y\left(\mathbf{x}^{(k)}\right)$. L'algorithme ainsi conçu considère une population initiale et un ensemble d'opérations de sélection, de croisement et de mutation. Il se déroule ensuite en trois phases : (1) tout d'abord, certains individus de la population sont choisis pour leurs performances (en fonction de la valeur de la fonction objectif qui leur est associée), (2) puis par croisement de ces individus, de nouveaux individus sont créés, et (3) finalement, une perturbation aléatoire des individus (mutation) permet de générer la population. L'algorithme continue itérativement jusqu'à convergence (régie par un critère d'arrêt). Parmi les différentes méthodes évolutionnaires, on distingue trois variantes [BACK, HAMMEL et SCHWEFEL 1997] : les algorithmes génétiques (*genetic algorithm - GA*) [HOLLAND 1962 ; HOLLAND 1975 ; GOLDBERG 1989]) les stratégies d'évolution (*evolution strategies - ES*) [RECHENBERG 1973 ; SCHWEFEL 1995] et la programmation évolutive (*evolutionary programming - EP*) [FOGEL 1962 ; FOGEL 1964].

- les **méthodes basées sur l'intelligence en essaims** sont fondées sur le comportement de certaines espèces animales. Plus précisément, certaines espèces qui vivent en colonie (de 2 ou plusieurs membres) adoptent un comportement de groupe pour assurer leur survie. Chaque individu (on parle également d'« agents ») du groupe semble avoir un comportement autonome mais celui-ci est au service du groupe, ce qui lui permet, par exemple, d'atteindre un endroit qu'un individu seul ne pourrait atteindre. L'étude et l'adaptation informatique de ces comportements a donné lieu à la création de nombreuses stratégies d'optimisation regroupées sous le terme générique d'intelligence en essaim (*swarm intelligence* [BONABEAU, DORIGO et THERAULAZ 1999]). Actuellement, de nombreux algorithmes basés sur le principe de l'intelligence en essaim sont développés. On se propose ici de se concentrer sur les deux principaux aujourd'hui utilisés :

  - les **algorithmes des colonies de fourmis** (*Ant Colony Optimization*) ont été proposés par DORIGO [DORIGO 1992 ; DORIGO et GAMBARDELLA 1997 ; DORIGO et DI CARO 1999]. Ces algorithmes ont été initialement introduits pour traiter des problèmes discrets tels que la recherche de chemins optimaux dans un graphe (par exemple le problème du voyageur de commerce, qui consiste à trouver le plus court chemin passant une seule fois par un ensemble de noeuds). Ils se basent sur le comportement des colonies de fourmis lorsque celles-ci cherchent de la nourriture. Lors de cette recherche, une fourmi parcourt au hasard l'espace autour de la colonie. Lorsqu'elle trouve une source de nourriture, elle en rapporte à la colonie tout en laissant sur son trajet des phéromones volatiles. Ces phéromones sont un marqueur que les autres fourmis ont tendance à suivre. Dans le cas où plusieurs sources de nourriture sont trouvées, la plus proche verra le trajet pour y accéder parcouru par le plus grand nombre de fourmis dans une même durée et, par conséquent, le dépôt de

phéromones sera plus important. À terme, les autres trajets seront délaissés (et le taux de phéromones sur ces trajets deviendra quasi-nul). Ce comportement de recherche de trajet le plus court a été mis en œuvre numériquement et progressivement amélioré afin de le rendre plus robuste [DORIGO et GAMBARDELLA 1997]. Il tient compte du phénomène de dépôt des phéromones, de leur volatilité (on parle d'évaporation) et du caractère aléatoire de la recherche. Pour cela, les algorithmes de colonie de fourmis intègrent un certains nombre de lois paramétriques permettant de restituer ces comportements.

Par la suite, le domaine d'application de cette famille d'algorithmes a été étendu aux problèmes d'optimisation continus [DRÉO 2004 ; SOCHA et DORIGO 2008].

- ▷ les **algorithmes par essaims particulaires** (*Particle Swarm Optimization*) ont été proposés par [J. KENNEDY et EBERHART 1995]. Ils mettent en œuvre numériquement une stratégie de collaboration entre agents. A l'instar de l'algorithme des colonies de fourmis, chaque agent n'a qu'une vision partielle du problème (dans la zone où il se déplace). L'algorithme pilote le comportement d'un ensemble de particules (formant l'essaim) en agissant sur leur position et leur vitesse. Il cherche itérativement à faire converger l'essaim vers la solution optimale du problème. Pour cela, l'algorithme calcule à chaque itération la vitesse de déplacement de chaque particule. Cette vitesse tient compte de plusieurs paramètres, des informations issues des autres particules de l'essaim et de la tendance d'évolution de la particule considérée. Ainsi, à l'instant $i+1$, la vitesse de la particule j (position $\mathbf{x}_{i+1}^{(j)}$) est calculée en tenant compte de : (1) la vitesse $\mathbf{v}_i^j$ de la particule à l'instant i, (2) la position de la particule $\mathbf{x}_b$ ayant la meilleure performance (associée à la fonction objectif la plus faible) parmi les particules voisines (on parle de comportement panurgien) et (3) la position $\mathbf{x}^{(j)*}$ où la particule considérée a atteint la meilleure performance. Ainsi la vitesse de la particule j à l'instant $i+1$ s'écrit $\mathbf{v}_{i+1}^j = c_1\mathbf{v}_i^j + c_2(\mathbf{x}^{(j)*} - \mathbf{x}_i^{(j)}) + c_3(\mathbf{x}_b - \mathbf{x}_i^{(j)})$ où c_1 est fixé arbitrairement et c_2 et c_3 sont choisis aléatoirement. La position de la particule j est ensuite calculée de la manière suivante $\mathbf{x}_{i+1}^{(j)} = \mathbf{x}_i^{(j)} + \mathbf{v}_{i+1}^j$. Il est possible de scinder l'essaim en sous-groupes et en adaptant le comportement panurgien à la nouvelle structure. Bien qu'ayant un principe de fonctionnement et une mise en œuvre assez aisés, ce type d'algorithme présente des points délicats tels que le choix des paramètres qui régissent le comportement des particules, le nombre de particules à choisir (pour avoir un bon compromis entre nombre d'itérations et convergence) et le type de scission de l'essaim. Quelques éléments complémentaires peuvent être trouvés dans [TRELEA 2003 ; CLERC 2005 ; PARSOPOULOS et VRAHATIS 2007 ; FONTAN 2011].

- ▶ la **recherche tabou** (*tabu search*) introduite par [GLOVER 1986 ; GLOVER et TAILLARD 1993]. Elle vise à améliorer les stratégies d'optimisation locale en y ajoutant un effet mémoire : les informations relatives aux minima déjà atteints ainsi

que les jeux de paramètres qui leur sont associés sont stockés au fur et à mesure des itérations. Ainsi la progression à chaque itération de l'algorithme se fait dans une direction et vers un jeu de paramètres qui n'est pas « tabou » (c'est-à-dire qui est autorisé). La mise en œuvre de cette démarche permet d'éviter l'arrêt de l'algorithme aux minima locaux.

- le **recuit simulé** se base sur un phénomène observé en metallurgie. La méthode de recuit permet d'améliorer les caractéristiques d'un matériau métallique en le chauffant graduellement puis en le refroidissant progressivement. Cette méthode est utilisée par exemple pour faciliter la relaxation des contraintes internes qui apparaissent au cœur des pièces suite à leur mise en forme. Elle permet de rétablir l'état d'équilibre de la micro-structure du matériau. Le recuit simulé vise à reproduire numériquement les phénomènes thermodynamiques qui se déroulent lors du recuit pour obtenir le nouvel état du matériau (obtenu en minimisant son énergie). Il a été développé par [KIRKPATRICK, GELATT et VECCHI 1983 ; ČERNÝ 1985] et emploie l'algorithme de Metropolis [METROPOLIS et al. 1953] qui permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le phénomène physique, l'algorithme de recuit simulé cherche à minimiser la fonction objectif (qui, par analogie avec le phénomène physique, serait l'énergie du système) par le biais d'un paramètre interne (qui serait la température). Cet algorithme assure une exploration à la fois aléatoire (l'algorithme de Metropolis cherche à reproduire le phénomène aléatoire de réagencement des atomes au sein du matériau basé sur l'équation de Boltzmann) et contrôlée afin de converger vers le minimum global en évitant les minima locaux (c'est ce qui se produit lorsque l'on refroidit de manière contrôlée la pièce pour atteindre l'état le plus stable à énergie minimale).

  Concrètement, à chaque itération, l'algorithme du recuit simulé évalue la valeur de la fonction objectif y en un ensemble de jeux de paramètres $\mathbf{x}^{(i)}$ ($i \in [\![1, n_p]\!]$). Pour chaque jeu de paramètres, si la valeur de la fonction objectif associée est inférieure à la plus faible valeur minimale atteinte auparavant, alors cette valeur et le jeu de paramètres associé sont retenus. Dans le cas contraire, le jeu de paramètres est retenu avec une probabilité $\exp(-\Delta y/T)$ (où Δy correspond à l'écart entre la meilleure valeur minimale de la fonction objectif déjà atteinte et celle associée au jeu de paramètres considéré). L'algorithme tire ensuite un nombre aléatoire compris entre 0 et 1. Si ce nombre est inférieur à la probabilité, alors le jeu de paramètres est retenu. Ainsi les jeux de paramètres pour lesquels la valeur de la fonction objectif n'est pas meilleure ne sont pas nécessairement rejetés. Cependant, la décroissance de la température T limite progressivement la liberté de recherche de l'algorithme, ce qui lui permet ainsi d'éviter un arrêt prématuré dans une zone de minimum local. Cette décroissance est pilotée itérativement par une loi (cf. [TRIKI, COLLETTE et SIARRY 2005]). Le recuit simulé est une méthode d'optimisation très populaire en raison de son efficacité et de sa capacité à s'adapter à de nombreux types de problèmes d'optimisation. Cependant, la convergence, bien qu'assurée sous certaines conditions (cf. [VAN LAARHOVEN et AARTS 1987]), est relativement lente et l'algorithme

nécessite de fixer empiriquement un nombre important de paramètres.

Ces méthodes dédiées à l'optimisation globale sont également classiquement regroupées sous le terme **meta-heuristique** [DRÉO 2004]. Ce terme désigne une méthode d'optimisation approchée capable de s'adapter facilement et sans grande modification de l'algorithme à un grand nombre de problèmes différents. Elle est, par ailleurs, utilisable en connaissant très peu d'informations concernant le problème. Elles s'appuient sur une base commune ayant les caractéristiques suivantes :

- elles ont un caractère stochastique qui assure une exploration du domaine de recherche ;
- ces méthodes sont directes (elles ne nécessitent pas le calcul du gradient) ;
- elles s'inspirent de comportements rencontrés dans la nature ;
- elles n'offrent aucune garantie sur l'optimalité de la meilleure solution fournie.

La figure 1.5 présente une classification des méta-heuristiques.

2.4 Approches multiniveaux

Les stratégies d'optimisation introduites précédemment requièrent un nombre important d'évaluations de la fonction objectif pour parvenir à localiser précisément le minimum global associé au problème d'optimisation considéré. Dans un contexte d'optimisation appliquée à un système complexe, dans lequel le coût nécessaire à l'obtention d'une réponse associée à un jeu de paramètres peut être prohibitif, il n'est pas envisageable d'employer de telles techniques. Afin de rendre les démarches d'optimisation qui nécessitent des appels à une fonction coûteuse accessibles, il est possible d'agir de trois manières :

- en améliorant les performances relatives au solveur fournissant les réponses associées aux jeux de paramètres. Cette approche, fortement intrusive, n'est pas toujours envisageable : l'accès aux sources du code n'est pas toujours possible (par exemple lors de l'usage de codes commerciaux) et les développements scientifiques et informatiques peuvent être extrêmement importants (par exemple pour la parallèlisation robuste d'un code).
- en améliorant les performances relatives à l'optimiseur. Les compétences nécessaires dans ce cas sont spécifiques à ce domaine et le caractère exploratoire de la recherche nécessaire à l'obtention du minimum global rend les méthodes existantes difficilement améliorables (en cherchant à réduire le nombre d'appels à la fonction pour assurer la convergence de l'algorithme).
- en améliorant la stratégie d'échanges entre le solveur et l'optimiseur (échanges indirects par exemple en utilisant un modèle intermédiaire).

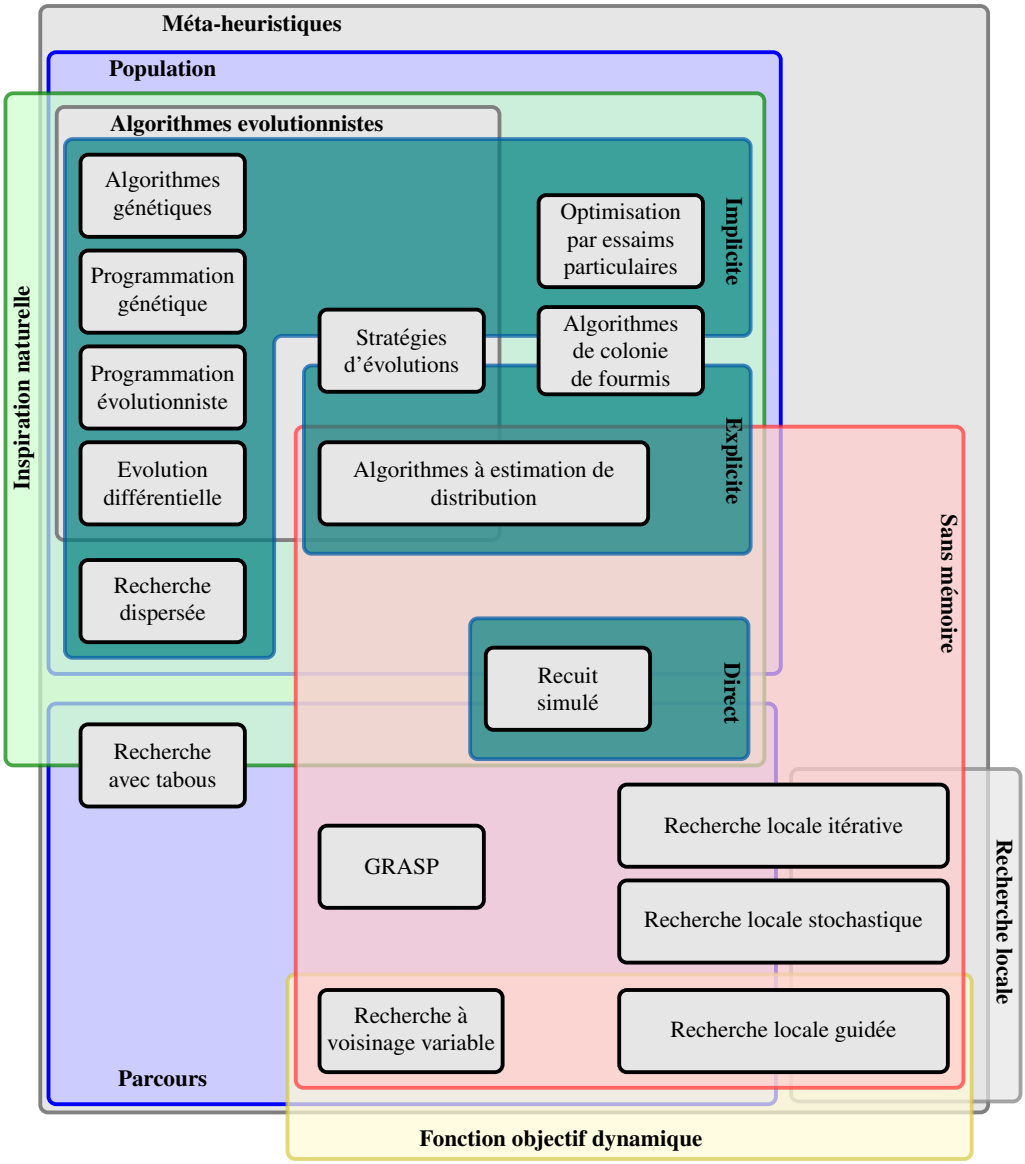


FIGURE 1.5 – Classification des algorithmes méta-heuristiques en fonction des caractéristiques de fonctionnement [DRÉO 2004]

Dans les travaux proposés ici, on s'intéresse au troisième point. L'idée maîtresse consiste à concevoir une stratégie assurant une optimisation progressive à coût maîtrisé. Pour cela, il est possible d'employer une approche multiniveaux [BOUCARD, BUYTET, SOULIER et al. 2010].

2.4.1 Vocabulaire et contexte

Largement employé lorsque l'optimisation se déroule en plusieurs étapes, le terme d'**optimisation multiniveaux** regroupe un ensemble de techniques d'optimisation proposant de scinder le processus d'optimisation complet en un ensemble de sous-processus. Ce type d'approche est fréquemment employé dans le cadre de l'optimisation multi-disciplinaire. Cette dernière cherche à mener à terme une optimisation faisant intervenir plusieurs disciplines distinctes. Une attention particulière doit être portée au degré de couplage existant entre les disciplines afin d'adapter au mieux la structure des décompositions multiniveaux et multi-disciplinaires retenues. En particulier, elles doivent assurer une communication pertinentes des grandeurs entre les niveaux et disciplines. [ENGELS, W. BECKER et A. MORRIS 2004] propose une distinction claire de ces deux stratégies d'optimisation (cf. figure 1.6) : les différentes disciplines mises en jeu sont traitées indépendamment en employant un code de calcul adapté à la problématique traitée. Par ailleurs, pour chaque discipline, des « sous-systèmes » forment chacun un niveau. À chaque niveau correspond un niveau de détails dans la discipline considérée. Dans la décomposition multiniveaux proposée, la stratégie consiste à construire un modèle de base plus ou moins grossier puis à construire des modèles de plus en plus fins afin de se concentrer sur une zone d'intérêt précise. La stratégie permet de se donner les moyens d'obtenir des informations de grande qualité dans les zones critiques (et non sur toute la structure, ce qui ne serait pas viable en termes de coût de calcul, de mémoire nécessaire...).

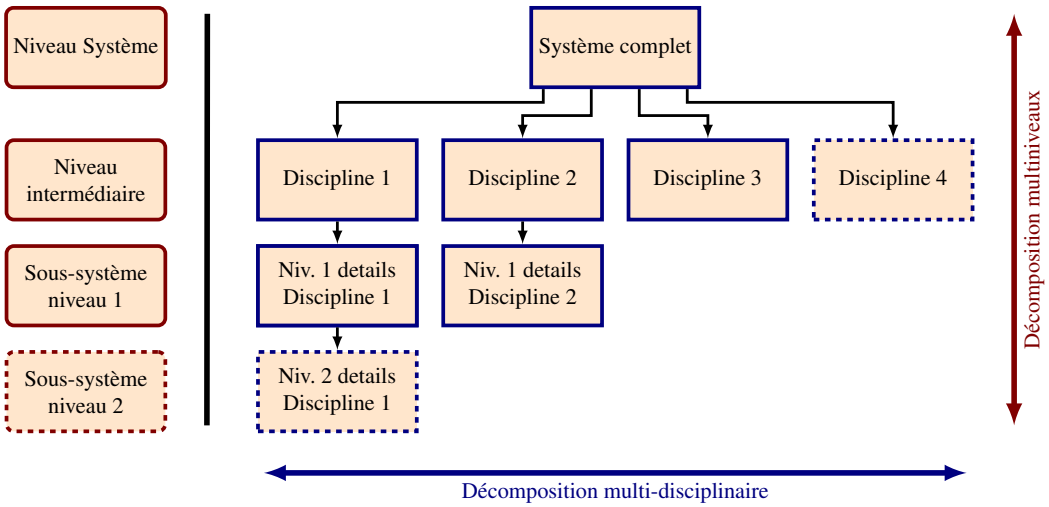


FIGURE 1.6 – Décomposition multiniveaux et multi-disciplinaire (schéma de principe adapté de [ENGELS, W. BECKER et A. MORRIS 2004])

Par le biais des deux décompositions proposées dans [ENGELS, W. BECKER et A. MORRIS 2004], le couplage entre les disciplines est très simple dans le sens où seul un nombre très limité de grandeurs provenant d'une discipline seront exploitées dans une autre. Le principal défaut de ce procédé est qu'il ne peut prendre en compte dans une discipline les modifications survenant dans une autre (comme par exemple les modifications locales de la géométrie, les changements d'état de surface...).

D'un point de vue optimisation, cette approche permet de traiter des problèmes d'optimisation distincts faisant intervenir des fonctions objectifs, fonctions contraintes et des variables de conception différentes à chaque niveau.

Dans certains travaux, cette séparation niveau/discipline n'est pas aussi claire et les auteurs considèrent chaque discipline comme un niveau de leur approche d'optimisation (voir par exemple [CHATTOPADHYAY, MCCARTHY et PAGALDIPTI 1995 ; T. CHEN et C. YANG 2005]).

Dans la suite de ce document, on ne considérera qu'une seule discipline. La démarche multiniveaux pourra être considérée de deux manières : une approche où chaque niveau consiste en une optimisation spécifique et une approche où chaque niveau s'appuie sur une modélisation différente de la problématique à traiter. On désignera respectivement ces techniques sous les noms **optimisation multiniveaux de paramètres** et **optimisation multiniveaux de modèles** [BOUCARD, BUYTET, SOULIER et al. 2010].

2.4.2 Optimisation multiniveaux de paramètres

Dans le cadre de ce type d'optimisation, le modèle mécanique et la fonction objectif associée sont définis dès le premier niveau de la stratégie et restent identiques tout le long du processus. L'idée de l'optimisation multiniveaux de paramètres réside dans la décomposition du problème d'optimisation : minimiser y sur $\mathcal{D}$ avec éventuellement prise en compte des contraintes d'égalités et/ou d'inégalités. Pour des raisons de clarté, dans la suite de ce paragraphe, nous ne tiendrons compte que d'un problème d'optimisation sans contrainte.

La décomposition consiste alors à transformer le problème :

$$\min_{(x_1,x_2,\ldots,x_{n_p})\in\mathcal{D}} y\big(x_1, x_2, \ldots, x_{n_p}\big) \tag{1.5}$$

en différents sous-problèmes formant chacun un niveau de l'approche multiniveaux :

$$\min_{(x_1^i,x_2^i,\ldots,x_{n_i}^i)\in\mathcal{D}} y\big(x_1, x_2, \ldots, x_{n_p}\big) \text{ avec } (x_1^1, \ldots, x_{n_{i-1}}^{i-1}, x_1^{i+1}, \ldots, x_{n_{i+1}}^{i+1}, \ldots, x_{n_p}) \text{ fixés} \tag{1.6}$$

où x_k^i ($k \in [\![1, n_i]\!]$) sont les n_i paramètres considérés au niveau i (parmi les n_l niveaux considérés). Cette renumérotation des paramètres permet de distinguer les paramètres associés à chaque niveau. Ainsi on a

$$(x_1^1, \ldots, \underbrace{x_1^i, \ldots, x_{n_i}^i}_{\text{niveau } i}, \underbrace{x_1^{i+1}, \ldots, x_{n_{i+1}}^{i+1}}_{\text{niveau } i+1}, \ldots, x_{n_{n_l}}^{n_l}) = (x_1, x_2, \ldots, x_{n_p})$$

L'idée simple de cette décomposition est de ne plus faire l'optimisation simultanément sur tous les paramètres mais de minimiser la fonction objectif seulement sur un nombre réduit de paramètres. Ainsi, lors du déroulement du processus d'optimisation, la liste des variables est progressivement enrichie. Il est alors possible de traiter le processus d'optimisation de deux manières différentes : par optimisation séquentielle ou par optimisation itérative.

■ **Optimisation séquentielle :** Elle consiste à réaliser à chaque niveau une optimisation sur l'ensemble des paramètres considérés à ce niveau : au niveau $i+1$, la liste des paramètres considérés du niveau i $(x_1^i, x_2^i, \ldots, x_{n_i}^i)$ est enrichie de nouveaux paramètres de manière à réaliser une optimisation sur un nouveau jeu de paramètres plus important $(x_1^i, x_2^i, \ldots, x_{n_i}^i, x_1^{i+1}, \ldots, x_{n_{i+1}}^{i+1})$. La figure 1.7 présente le principe de l'optimisation séquentielle.

Cette méthode s'appuie sur un classement par ordre d'importance des paramètres. Il peut être obtenu en considérant tout d'abord les plus globales et en y ajoutant progressivement les variables les plus locales. Il est également possible de classer les variables des plus influentes aux moins influentes en utilisant une étude de sensibilité.

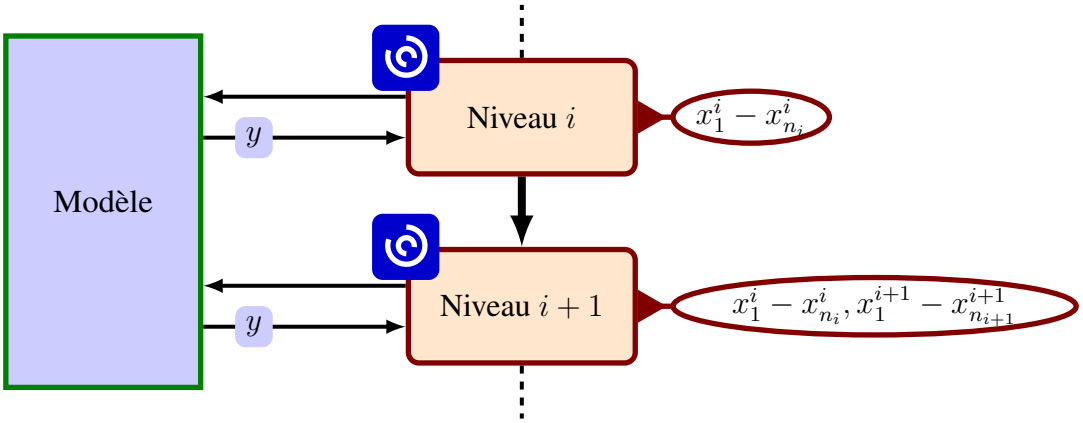


FIGURE 1.7 – Principe de l'optimisation séquentielle multiniveaux de paramètres, illustration sur 2 niveaux

Quelques illustrations de cette approche peuvent être trouvées dans [S. KRAVANJA, SORŠAK et Z. KRAVANJA 2003 ; B. LIU, HAFTKA et L. T. WATSON 2004 ; S. KRAVANJA, ŠILIH et Z. KRAVANJA 2005] qui présentent des approches d'optimisation à deux niveaux incorporant de nombreux types de paramètres afin d'optimiser des structures (treillis ou poutres) pour assurer des réductions de masses, de coûts...

■ **Optimisation itérative :** Cette approche itérative consiste, au niveau i, à maintenir fixées les valeurs optimales des paramètres x_k^{m*} (avec $m = \{1, 2, \ldots, i-1, i+1, n_l\}$ et $k = \{1, \ldots, n_m\}$) obtenues aux niveaux précédents et à réaliser l'optimisation en tenant compte des paramètres $(x_1^i, \ldots, x_{n_i}^i)$ comme variables d'optimisation. Ainsi, au cours du processus, les optimisations des différents niveaux sont exécutées itérativement en tenant compte

des valeurs optimales des paramètres obtenues dans les autres niveaux. On considère ici qu'une itération correspond à un cycle d'optimisation portant sur tous les niveaux de la décomposition multiniveaux. Le jeu de paramètres optimal obtenu à convergence de la stratégie d'optimisation itérative multiniveaux de paramètres $(x_1^*, \ldots, x_{n_p}^*)$ sera construit itérativement et à chaque niveau (à chaque itération et à chaque niveau le jeu de paramètres optimal pourra être modifié). La figure 1.8 illustre ce principe.

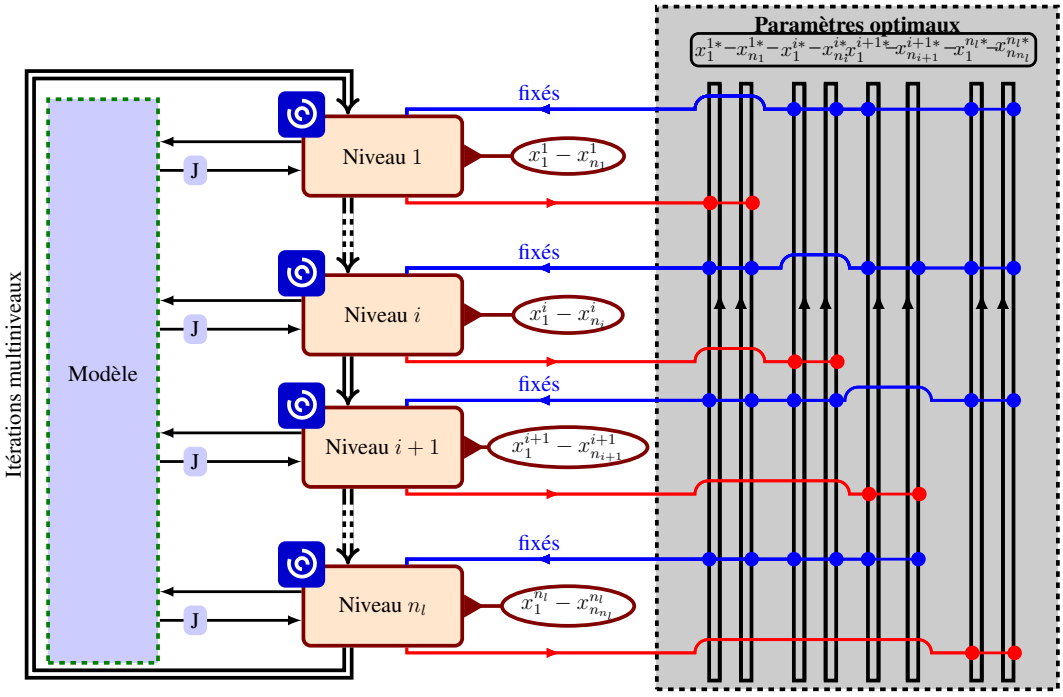


FIGURE 1.8 – Principe de l'optimisation itérative multiniveaux de paramètres

Un exemple d'application de cette stratégie peut être trouvé dans [T. CHEN et C. YANG 2005]. Dans la situation où les optimisations de chaque niveau seraient totalement découplées (c'est-à-dire si les variables d'optimisation prises en compte dans chaque niveau n'aurait pas d'impact sur les optimisations des autres niveaux), le processus itératif ne serait alors pas nécessaire. Dans le cadre des travaux de [LE RICHE et GAUDIN 1998], cette approche sans itération est mise en œuvre du fait de l'indépendance des comportement dans le plan et hors plan d'un matériau composite.

Dans la majorité des cas, le processus itératif est donc nécessaire et permet de tenir compte du couplage existant entre les différents niveaux. Des exemples de mise en œuvre de la stratégie itérative peuvent être trouvés dans les travaux [BENDSØE 1995 ; THEOCARIS et STAVROULAKIS 1998] portant sur l'optimisation de matériaux composites en tenant compte de leur micro-structure ou dans les travaux [CONCEIÇÃO ANTÓNIO 2002] portant sur l'optimisation de matériaux composites en prenant en compte des non-linéarités géométriques.

Dans certains cas, les stratégies d'optimisation multiniveaux de paramètres séquentielle ou itérative peuvent ne pas converger vers la solution optimale du problème initial (voir

[ALEXANDROV et R. M. LEWIS 2000]) mais vers une solution optimale directement liée à la décomposition retenue et aux différentes approches numériques mises en œuvre dans chaque niveau (par exemple les algorithmes d'optimisation choisis). Par ailleurs, la présence de contraintes dans le problème d'optimisation initial peut rendre la décomposition particulièrement ardue et entraîner également des problèmes de convergence.

2.4.3 Optimisation multiniveaux de modèles

À la différence des méthodes présentées dans la partie 2.4.2 où la décomposition du processus d'optimisation consistait à scinder les jeux de paramètres en « sous-jeux » de paramètres et à réaliser les optimisations à chaque niveau sur un même modèle, l'approche proposée dans cette partie s'appuie sur une décomposition du modèle initial en « sous-modèles ». Chaque niveau de cette approche sera donc construit autour d'un des modèles de la décomposition. Les stratégies d'optimisation multiniveaux de modèles peuvent être regroupées en deux groupes selon les liens et dépendances existant entre les modèles. On parle d'optimisation multiniveaux de modèles hiérarchiques et d'optimisation multiniveaux de modèles imbriqués.

■ **Optimisation hiérarchique :** Dans le cadre de ce type d'optimisation, l'ensemble des niveaux et par conséquent des modèles est construit de manière à tenir compte de l'interdépendance des modèles. Ainsi, une véritable hiérarchie des modèles est élaborée qui permet par exemple de partir d'un modèle grossier et de le raffiner progressivement. Les informations acquises au niveau grossier lors du processus d'optimisation, qui lui est associé, sont ensuite transmises au modèle suivant. Ces informations peuvent être de deux types : d'une part, les valeurs des paramètres solutions du problème d'optimisation réalisé sur le modèle grossier et, d'autre part, un ensemble d'informations complémentaires qui peuvent être prises en compte par le modèle fin (par exemple la nécessité de raffiner une zone en particulier) de manière directe ou indirecte (nécessitant par exemple un calcul complémentaire). La figure 1.9 présente schématiquement le principe de fonctionnement de ce type d'optimisation. Dans le cadre de ce type d'optimisation, il n'y a pas de restrictions sur les paramètres d'optimisation, c'est-à-dire que les paramètres du niveau $i+1$ ne sont pas nécessairement ceux du niveau i.

Le plus souvent, à chaque niveau, l'optimisation est réalisée de manière totalement indépendante des autres niveaux : aucune information n'est communiquée au niveau inférieur (ce qui n'est pas problématique si les niveaux inférieurs ont peu d'influence sur les niveaux supérieurs). Les modèles des niveaux inférieurs sont alors complètement oubliés.

Cette approche d'optimisation multiniveaux de modèles hiérarchiques est mise en œuvre dans [ROBINSON et KEANE 1999 ; KEANE et PETRUZZELLI 2000] dans le cadre de la minimisation de la traînée d'une aile d'avion. Pour cette étude, trois modèles de plus en plus riches sont employés : le degré finesse se base sur le maillage éléments finis et sur le modèle de comportement du fluide. Pour ces trois niveaux, les mêmes paramètres d'optimisation (paramètres géométriques pilotant la forme de l'aile) sont pris en compte à

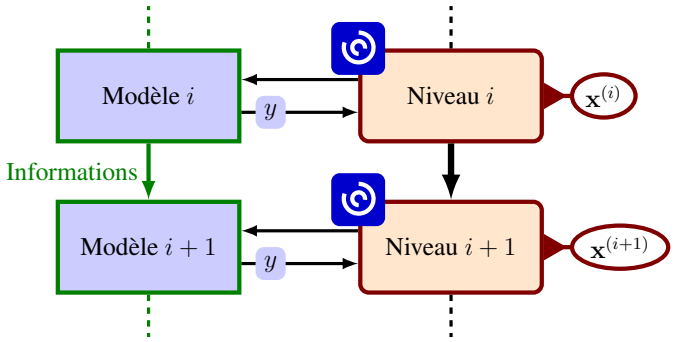


FIGURE 1.9 – Principe de l'optimisation multiniveaux de modèles hiérarchiques, illustration sur 2 niveaux

chaque niveau.

■ **Optimisation imbriquée :** L'idée est ici de procéder à une décomposition du processus d'optimisation d'une manière similaire à l'approche hiérarchique : à chaque augmentation de niveau, la qualité du modèle est améliorée. La différence réside ici dans le fait que les modèles communiquent entre eux dans les deux sens. En assurant un passage d'informations bi-directionnel, il est possible de restituer une partie du couplage inter-niveaux. Cependant, cette communication ne peut se faire qu'à la condition d'avoir des modèles sachant communiquer entre eux, ce qui peut nécessiter parfois quelques manipulations numériques pour rendre les données compréhensibles entre les niveaux. La figure 1.10 présente schématiquement le principe de fonctionnement de ce type d'optimisation.

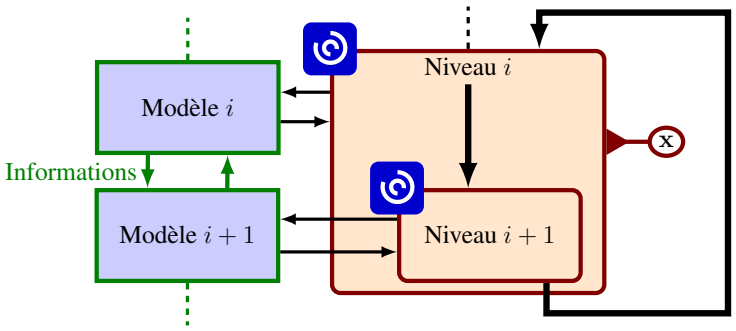


FIGURE 1.10 – Principe de l'optimisation multiniveaux de modèles imbriqués

Cette approche imbriquée pourrait être mise en œuvre dans le cadre d'une description multi-échelles d'un problème. Cette description cherche à modéliser en parallèle les différentes échelles d'une même structure. Par le biais de cette description, il est alors possible de tenir compte de l'influence de l'optimisation à l'échelle fine sur celle à l'échelle grossière. Un exemple d'approche imbriquée peut être trouvé dans [ENGELS, W. BECKER et A. MORRIS 2004] dans le cadre d'une optimisation multi-disciplinaire. Au sein de la discipline « structure », une optimisation imbriquée est réalisée. Elle vise à minimiser la masse totale d'une aile d'avion sous la contrainte d'inégalité basée sur une limite des

contraintes de Von Mises à ne pas dépasser. Les deux modèles employés sont un modèle d'aile d'avion utilisant la méthode des éléments finis et un super-élément assurant une description fine d'une zone d'intérêt. La stratégie d'optimisation proposée tient compte et assure le couplage entre les deux modèles.

3 Calculs d'assemblages pour l'optimisation

L'emploi de méthodes d'optimisation globale peut nécessiter un nombre important de simulations pour converger vers un optimum. Leur application dans le cadre de l'optimisation d'assemblages nécessite de s'adapter à deux problématiques *a priori* distinctes :

- la première repose sur la capacité de construire un **modèle mécanique représentatif des assemblages** considérés. Il doit être capable de prendre en compte la présence de plusieurs pièces et d'interfaces parfois nombreuses entre celles-ci. La réelle difficulté consiste ici à modéliser les phénomènes physiques complexes et non-linéaires pouvant survenir au niveau des interfaces. En particulier, la prise en compte du frottement et du contact sera indispensable.

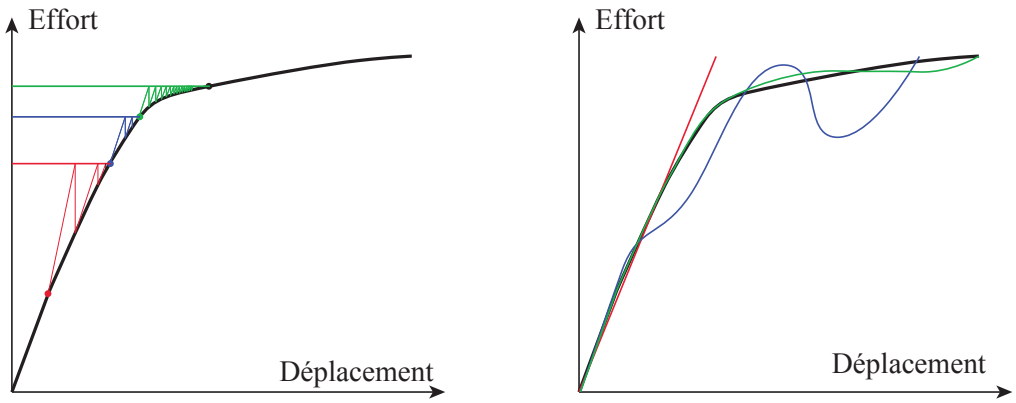
- la seconde est une problématique intrinsèque à l'optimisation globale et aux études paramétriques : ces méthodes requièrent un **nombre important de résolutions similaires** (dans le sens où seuls quelques paramètres varient). Le temps de calcul qui en découle peut alors devenir extrêmement important.

Dans cette partie, des outils relatifs aux deux problématiques seront présentés. Dans un premier temps, le principe de la méthode LATIN pour les assemblages sera rappelé. Puis quelques stratégies dédiées aux études paramétriques ou à l'optimisation seront ensuite évoquées. L'accent sera porté sur la stratégie multiparamétrique que l'on emploiera dans le cadre de ces travaux de thèse.

3.1 La méthode LATIN

Introduite dans les années 1980 par P. Ladevèze, la méthode LATIN [5] (*Large Time Increment*) permet de résoudre des problèmes mécaniques non-linéaires [LADEVÈZE 1985]. Contrairement à bon nombre de méthodes permettant de résoudre ce type de problèmes, pour lesquelles il est nécessaire de procéder par incrément de temps pour obtenir la solution du problème et son évolution temporelle, la méthode LATIN propose de résoudre le problème de façon globale en temps. À chaque itération de l'algorithme, le problème est résolu sur tout l'intervalle de temps considéré et en tout point de la structure. La solution est ensuite enrichie à chaque itération jusqu'à convergence de l'algorithme. La figure 1.11 présente schématiquement le comportement d'une méthode incrémentale et de la méthode LATIN.

5. Classiquement prononcé [latin]



(a) Méthode incrémentale **(b)** Méthode LATIN

FIGURE 1.11 – Représentation schématique du comportement de deux stratégies de résolution

L'interruption volontaire ou involontaire de l'algorithme de résolution conduit :

- dans le cas d'une stratégie incrémentale, à obtenir la solution **exacte** jusqu'au pas de temps $t_i < T$ (où t_i correspond au dernier incrément de temps calculé) ;
- dans le cas de l'approche LATIN, à obtenir une solution **approchée** de la solution sur l'ensemble de l'intervalle de temps considéré (sur $[0, T]$).

La méthode LATIN se base sur 3 principes [LADEVÈZE 1999] :

- **P1** : une séparation des difficultés consistant à former deux groupes d'équations (cf. partie 3.1.3 page 46) ;
- **P2** : la résolution par un schéma itératif à deux étapes (cf. partie 3.1.4 page 48) ;
- **P3** : le choix d'une représentation des inconnues adaptées au cadre de l'étude (cf. partie 3.1.7 page 52).

Par ailleurs, dans le cadre du calcul de structures assemblées, une décomposition de domaine décrite dans la partie 3.1.2 page 44 sera considérée.

Remarque 1.2 : Dans le cadre de ces travaux, l'approche employée ne sera basée que sur les points **P1** et **P2**. Comme nous le verrons brièvement dans la partie 3.1.7 page 52, le point **P3** ajoute une information à fort caractère mécanique permettant d'accélérer la convergence de l'algorithme de résolution. N'étant pas disponible dans les outils employés, il ne fut pas utilisé.

La méthode LATIN a été développée au LMT-Cachan et employée dans de nombreux cas de figure. Le tableau 1.1 rappelle les principaux développements ayant été réalisés autour de la méthode LATIN.

Dans le cadre de la simulation d'assemblage, on pourra également se tourner vers [LADEVÈZE 1990 ; BLANZÉ, DANWÉ et al. 1992 ; BLANZÉ, CHAMPANEY, COGNARD et al. 1996 ; CHAMPANEY 1996] pour plus de détails.

		Ref.	PGD
Grandes déformations et plasticité		[VAUCHEZ 1991]	✓
		[B. LIU 1992]	✓
Grandes transformations, formulation corotationnelle		[BOUCARD 1996]	✓
		[MICHEL-PONNELLE 2001]	✓
Élasto-visco-plasticité, grand nombre de cycles et thermo-élasticité		[BOISSE 1987]	✓
		[COGNARD 1989]	✓
		[ARZT 1994]	✓
		[COGNARD, LADEVÈZE et TALBOT 1999]	✓
Vérification visco-plasticité, PGD		[PELLE et RYCKELYNCK 2000]	✓
Dynamique non-linéaire		[ROYER 1990]	✓
Endommagement & composites	Formulation fonctionnelle (loi de comportement intégrale)	[ALLIX 1992] [GUINARD et al. 2002]	✓
	Formulation variables internes	[DOUCHIN 2000]	
	Multi-échelles composite	[TROVALET 2010]	
	Couplage flambage-délaminage	[SAAVEDRA REDLICH 2012]	
Dynamique rapide, chocs pyrotechniques (DDM)		[LEMOUSSU 2000] [DERUMAUX 2004] [SEN GUPTA 2005]	
Assemblages	2D-axisymétrique, statique	[DANWE 1993]	
	3D (brides en statique)	[CHAMPANEY 1996]	
	Applications, quasi-statique	[BLANZÉ, CHAMPANEY et VEDRINE 2000]	
	Dynamique, chocs pyro.	[LEMOUSSU 2000]	
	Optimisation	[BOUCARD, BUYTET et GUIDAULT 2007] [SOULIER 2007]	
DDM	Parallèlisme	[LADEVÈZE 1987] [LORONG 1994] [DUREISSEIX 1997]	
	Multi-échelles	[LADEVÈZE et DUREISSEIX 1999] [LOISEAU 2001]	
	Stratégie réutilisation, stochastique	[NOUY 2003] [LOISEAU 2001]	
	Multiphysiques	[NÉRON 2004]	
	Couplage XFem, fissuration	[GUIDAULT 2005]	
	Multi-échelles 3D, code C++, LMTpp	[VIOLEAU 2007]	
	Multi-échelles en temps	[PASSIEUX 2008]	✓
	Composites	[KERFRIDEN 2008]	
	Amortissements	[CAIGNOT 2009]	
Multiparamétrique	Assemblages	[BOUCARD 2001] [CHAMPANEY et BOUCARD 2003]	
	Dynamique multi-échelles	[ODIÉVRE 2009]	
	Assemblages de composites	[ROULET 2011]	
	Flambage, élasto-visco-plasticité	[BOUCARD et LADEVÈZE 1999]	✓
	Élasto-visco-plasticité	[RELUN 2011]	✓
	Base optimale	[HEYBERGER, BOUCARD et NÉRON 2012]	✓
Identification Problèmes inverses		[ALLIX et VIDAL 2002] [NGUYEN, ALLIX et FEISSEL 2008]	✓

TABLE 1.1 – Les grands développements autour de la méthode LATIN

Stratégie multiparamétrique et métamodèles pour l'optimisation multiniveaux de structures

3.1.1 Hypothèses, notations et problème de référence

Dans la suite de ce document, nous restreindrons notre étude en considérant les **pièces** de l'assemblage comme ayant un **comportement linéaire élastique isotherme** sous l'hypothèse des petites perturbations dans le cadre des chargements quasi-statiques dans l'intervalle de temps $[0, T]$. Les **non-linéarités** considérées seront donc situées uniquement au niveau des **interfaces** entre les pièces. Par ailleurs, la stratégie employée pour simuler les assemblages sera basée sur la méthode des éléments finis.

■ **Définition du problème de référence :** On considère un domaine Ω soumis à des forces volumiques $\mathbf{f}^{\mathrm{d}}$ et à des efforts $\mathbf{F}^{\mathrm{d}}$ sur une partie $\partial_2\Omega$ de sa frontière (cf. figure 1.12). La partie de la frontière $\partial_1\Omega$ (telle que $\partial_1\Omega \cap \partial_2\Omega = \varnothing$) est soumise à des déplacements imposés $\mathbf{u}^{\mathrm{d}}$.

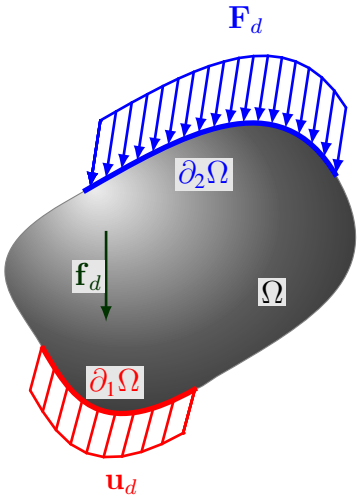


FIGURE 1.12 – Problème de référence

■ **Notations :** On définit les espaces vectoriels $\mathcal{U}_{ad}^{[0,T]}$ et $\mathcal{S}_{ad}^{[0,T]}$ tels que

► Soit M un point de Ω, un champ de déplacement $\mathbf{u} = \mathbf{u}(M, t)$ est dit **cinématiquement admissible** (CA) s'il est *suffisamment régulier* et s'il vérifie les liaisons (c'est-à-dire que les déplacements sur le bord $\partial_1\Omega$ sont égaux à $\mathbf{u}^{\mathrm{d}}$). L'espace vectoriel $\mathcal{U}_{ad}^{[0,T]}$ des champs des déplacements cinématiquement admissibles est donc défini tel que :

$$\mathcal{U}_{ad}^{[0,T]} = \Big\{ \mathbf{u}(M,t) | \mathbf{u} \text{ est } \textit{suffisamment régulier}, \quad \forall t \in [0,T],\ \forall M \in \partial_1\Omega,\ \mathbf{u}(M,t) = \mathbf{u}_d \Big\} \quad (1.7)$$

De plus, on définit l'espace vectoriel $\mathcal{U}_{ad,0}^{[0,T]}$ des champs de déplacements *cinématiquement admissibles à zéro* (CA0) tel que :

$$\mathcal{U}_{ad,0}^{[0,T]} = \Big\{ \mathbf{u}(M,t) | \mathbf{u} \text{ est } \textit{suffisamment régulier}, \\ \forall t \in [0,T],\ \forall M \in \partial_1\Omega,\ \mathbf{u}(M,t) = \mathbf{0} \Big\} \quad (1.8)$$

Ainsi un champ de déplacement $\mathbf{u} = \mathbf{u}(M,t)$ est dit cinématiquement admissible à zéro s'il est *suffisamment régulier* et s'il vérifie des conditions de déplacement nul sur le bord $\partial_1\Omega$.

- Soit M un point de Ω, un champ de contraintes $\boldsymbol{\sigma} = \boldsymbol{\sigma}(M,t)$ est dit **statiquement admissible** (SA) s'il est *suffisamment régulier*, *symétrique* et s'il vérifie les équations d'équilibre. L'espace vectoriel $\mathcal{S}_{ad}^{[0,T]}$ des champs des contraintes statiquement admissibles est donc défini tel que :

$$\mathcal{S}_{ad}^{[0,T]} = \Big\{ \boldsymbol{\sigma}(M,t) | \boldsymbol{\sigma} \text{ est } \textit{suffisamment régulier, symétrique}, \\ \forall \mathbf{u}^* \in \mathcal{U}^0,\ \int_\Omega \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathbf{u}^*) d\Omega = \int_\Omega \mathbf{f}_d \cdot \mathbf{u}^* d\Omega + \int_{\partial_2\Omega} \mathbf{F}_d \cdot \mathbf{u}^* dS \Big\} \quad (1.9)$$

où $\mathcal{U}^0 \subset \mathcal{U}_{ad,0}^{[0,T]}$ et $\boldsymbol{\varepsilon}(\mathbf{u}^*)$ représente le champ de déformation engendré par le champ de déplacement $\mathbf{u}^*$. Sous l'hypothèse des petites perturbations retenues ici,

$$\forall M \in \Omega,\ \forall t \in [0,T],\ \boldsymbol{\varepsilon}(M,t) = \boldsymbol{\varepsilon}\left(\mathbf{u}(M,t)\right) \\ = \frac{1}{2}\left(\left(\frac{\mathrm{d}\mathbf{u}(M,t)}{\mathrm{d}\mathbf{OM}} \right)^T + \frac{\mathrm{d}\mathbf{u}(M,t)}{\mathrm{d}\mathbf{OM}} \right) \quad (1.10)$$

On définit également l'espace vectoriel $\mathcal{S}_{ad,0}^{[0,T]}$ des champs de contraintes *statiquement admissibles à zéro* (SA0) tel que :

$$\mathcal{S}_{ad,0}^{[0,T]} = \Big\{ \boldsymbol{\sigma}(M,t) | \boldsymbol{\sigma} \text{ est } \textit{suffisamment régulier, symétrique}, \\ \forall \mathbf{u}^* \in \mathcal{U}^0,\ \int_\Omega \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathbf{u}^*) d\Omega = 0 \Big\} \quad (1.11)$$

■ **Formulation du problème de référence :**

Problème 1.4
Trouver $(\mathbf{u}(M,t), \boldsymbol{\sigma}(M,t))$ définis sur $\Omega \times [0,T]$ tels que :

- $\mathbf{u}(M,t) \in \mathcal{U}_{ad}^{[0,T]}$ ($\mathbf{u}(M,t)$ CA)
- $\boldsymbol{\sigma}(M,t) \in \mathcal{S}_{ad}^{[0,T]}$ ($\boldsymbol{\sigma}(M,t)$ SA)
- $(\mathbf{u}(M,t), \boldsymbol{\sigma}(M,t))$ vérifie la relation de comportement : $\boldsymbol{\sigma}(M,t) = \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u}(M,t))$ où $\mathbf{D}$ est l'opérateur de Hooke.

Remarque 1.3 : Le problème de référence introduit ci-dessus correspond à un problème d'élasticité linéaire sous l'hypothèse des petites perturbations. Les non-linéarités présentes dans le cadre des assemblages n'y apparaissent pas. En effet, ces dernières apparaissent aux niveaux des interfaces entre les sous-domaines et seront introduites suite à l'étape de décomposition de domaine faisant l'objet de la section suivante.

Remarque 1.4 : Afin de ne pas surcharger les notations dans la suite de ce document, celles-ci seront allégées : une grandeur notée jusqu'à présent $\bullet(M,t)$ sera notée simplement $\bullet$ (la dépendance aux variables temporelles et d'espace sera sous-entendue).

3.1.2 Décomposition de domaine

Dans le cas de modélisations tridimensionnelles de structures complexes, la complexité de la géométrie et la recherche d'une solution de bonne qualité conduisent à l'augmentation du nombre de degrés de liberté. Les systèmes linéaires associés deviennent alors très importants.

C'est pour cette raison qu'il est nécessaire d'employer une méthode de décomposition de domaine conduisant à la résolution de problèmes de taille réduite et à la possibilité de paralléliser la résolution. La décomposition conduit alors à considérer des **sous-domaines** et des **interfaces**. Cette décomposition, dans le cas des assemblages, est relativement naturelle car chaque pièce peut être considérée comme un sous-domaine et chaque zone de contact entre deux pièces comme une interface.

Parmi les méthodes classiques de décomposition de domaines, on distingue les approches avec recouvrement éventuel des sous-domaines [SCHWARZ 1869] et celles sans recouvrement, parmi lesquelles on retrouve :

- les **approches primales** vérifiant la continuité des déplacements aux interfaces, telles que la méthode de Schur primale [PRZEMIENIECKI 1963] et la méthode BDD (*Balancing Domain Decomposition Method*) [MANDEL 1993] ;
- les **approches duales** assurant l'équilibre des efforts aux interfaces, telles que la méthode de Schur Duale et la méthode FETI (*Finite Element Tearing and Interconnecting*) [FARHAT et ROUX 1991] ;

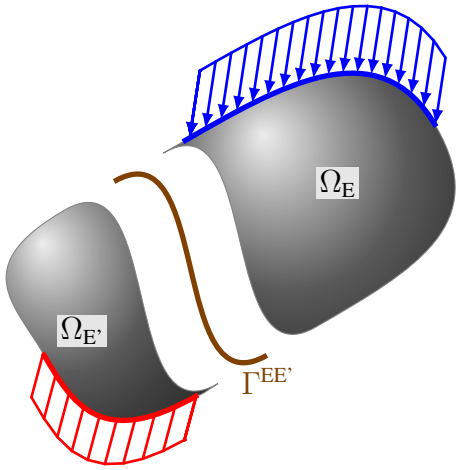


FIGURE 1.13 – Principe de la décomposition de domaine sans recouvrement en sous-domaines et interfaces

- les **approches mixtes** raccordant les efforts et déplacements aux interfaces, telles que les méthodes basées sur un algorithme de type Lagrangien augmenté [GLOWINSKI et LE TALLEC 1990], la méthode LaTIn [LADEVÈZE 1985], la méthode FETI Dual-Primal [FARHAT, LESOINNE et PIERSON 2000] et la méthode FETI à deux champs (ou à deux multiplicateurs de Lagrange) [SERIES, FEYEL et ROUX 2003].

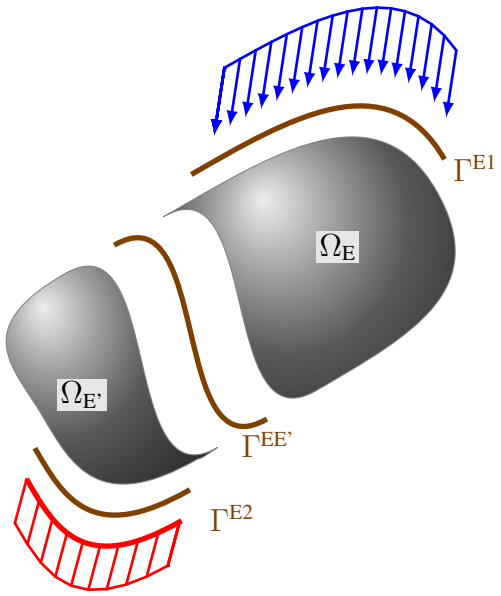
Une review détaillée portant sur les méthodes de décomposition de domaine sans recouvrement peut être trouvée dans [GOSSELET et REY 2006].

Dans le cadre de l'approche considérée ici, la méthode proposée se base sur l'emploi d'une méthode de décomposition de domaines mixte qui présente la particularité de considérer les sous-domaines et les interfaces comme des entités mécaniques de la structure et/ou de l'assemblage à **part entière** [LADEVÈZE 1985 ; LADEVÈZE 1999]. Cette particularité permet de rendre plus aisé le traitement des comportements d'interface tels que le contact et le frottement. De manière générale, les conditions aux limites sont prises en compte par l'intermédiaire d'interfaces entre les sous-domaines et l'extérieur.

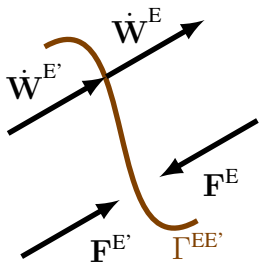
Ω_E et $\Gamma^{EE'}$ désignent respectivement le sous-domaine E et l'interface entre les sous-domaines Ω_E et $\Omega_{E'}$.

Au niveau des interfaces, sont considérés les champs d'effort $\mathbf{F}^E$ et $\mathbf{F}^{E'}$ correspondant aux efforts exercés par l'interface sur les deux sous-domaines et les variables duales que sont, dans le cas statique, les déplacements $\mathbf{W}^E$ et $\mathbf{W}^{E'}$ et dans le cas quasi-statique, les vitesses $\dot{\mathbf{W}}^E$ et $\dot{\mathbf{W}}^{E'}$. La figure 1.14 présente la décomposition employée dans la méthode LATIN.

Dans le cadre proposé, l'hypothèse de comportement linéaire élastique sous l'hypothèse des petites perturbations considérée pour les sous-domaines permet d'assurer la complète définition du problème par l'intermédiaire des inconnues d'interfaces. Ainsi la solution du problème sous-structuré s'écrit $\mathbf{s} = (\mathbf{W}, \mathbf{F})$ union des inconnues d'interface $\mathbf{s}^E$ définies en



(a) Décomposition de l'approche LATIN



(b) Grandeurs d'interface (déplacements et efforts imposés à l'interface)

FIGURE 1.14 – Décomposition prise en compte dans la méthode LATIN et grandeurs aux interfaces

tout point $M \in \partial_E\Omega$ (sur le bord de Ω_E) et en tout instant $t \in [0,T]$:

$$\mathbf{s} = \bigcup_E \mathbf{s}^E = \bigcup_E (\mathbf{W}^E, \mathbf{F}^E) \tag{1.12}$$

Remarque 1.5 : Les espaces vectoriels introduits pour le problème de référence (partie 3.1.1 page 42) peuvent être adaptés à chaque sous-domaine Ω_E en tenant compte des déplacements $\mathbf{W}^E$ et efforts $\mathbf{F}^E$ d'interface s'exerçant sur leurs frontières. Les sous-espaces associés à chaque sous-domaine Ω_E seront notés $\mathcal{U}_{E,ad}^{[0,T]}$ et $\mathcal{S}_{E,ad}^{[0,T]}$.

3.1.3 Séparation des difficultés

Le premier point de l'approche LATIN (point **P1**) consiste à séparer les difficultés. De manière générale, l'objectif est ici de former deux groupes d'équations permettant d'éviter de traiter un problème qui soit à la fois global et non-linéaire. On forme alors deux sous-espaces notés : $\mathbf{A_d}$ et $\boldsymbol{\Gamma}$. Ces deux sous-espaces contiennent respectivement les solutions aux **équations linéaires** (éventuellement globales) et les solutions aux **équations locales** en variables d'espace (éventuellement non-linéaires).

Dans le cas de la simulation des assemblages, et sous les hypothèses retenues (cf. partie 3.1.1 page 42), les non-linéarités sont uniquement présentes au niveau des interfaces.

Ainsi la décomposition en deux groupes d'équations revient, ici, à considérer un problème dans les sous-domaines ($\mathbf{A_d}$) et un problème aux interfaces ($\mathbf{\Gamma}$).

■ **Problème dans les sous-structures :** On définit $\mathbf{u}^{\mathrm{E}}$ (M,t) les champs de déplacement en tout point M du sous-domaine Ω_{E} et en tout instant $t \in [0,T]$. Soit $\boldsymbol{\sigma}^E$ et $\boldsymbol{\varepsilon}^E$ les champs de contrainte et de déformation associés.

Le problème dans un sous-domaine Ω_{E} revient à trouver l'histoire des champs des déplacements et des contraintes solutions du sous-espace $\mathbf{A_d}$ regroupant :

- l'admissibilité cinématique (conditions de liaisons sur les bords du sous-domaine Ω_{E}) $\mathbf{u}^{\mathrm{E}} \in \mathcal{U}_{E,ad}^{[0,T]}$:

$$\forall M \in \partial\Omega_{\mathrm{E}} \text{ et } \forall t \in [0,T], \quad \mathbf{u}^{\mathrm{E}}(M,t) = \mathbf{W}^{\mathrm{E}}(M,t) \tag{1.13}$$

- l'admissibilité statique (traduisant l'équilibre du sous-domaine Ω_{E}) $\boldsymbol{\sigma}^E \in \mathcal{S}_{E,ad}^{[0,T]}$:

$$\forall \mathbf{u}^* \in \mathcal{U}^0 \text{ et } \forall t \in [0,T]$$
$$\int_{\Omega_{\mathrm{E}}} \boldsymbol{\sigma}^E : \boldsymbol{\varepsilon}(\mathbf{u}^*) d\Omega = \int_{\Omega_{\mathrm{E}}} \mathbf{f}_d \cdot \mathbf{u}^* d\Omega - \int_{\partial\Omega_{\mathrm{E}}} \mathbf{F}^{\mathrm{E}} \cdot \mathbf{u}^* dS \tag{1.14}$$

- la relation de comportement (comportement linéaire élastique du sous-domaine Ω_{E})

$$\forall M \in \Omega_{\mathrm{E}} \text{ et } \forall t \in [0,T], \quad \boldsymbol{\sigma}^E = \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u}^{\mathrm{E}}) \tag{1.15}$$

où $\mathbf{D}$ est l'opérateur de Hooke.

■ **Problème aux interfaces :** Le problème dans une interface $\Gamma^{\mathrm{EE'}}$ revient à déterminer l'histoire des champs de forces $\mathbf{F}^{\mathrm{E}}(M,t)$ et $\mathbf{F}^{\mathrm{E'}}(M,t)$ et des champs de vitesse $\dot{\mathbf{W}}^{\mathrm{E}}(M,t)$ et $\dot{\mathbf{W}}^{\mathrm{E'}}(M,t)$ avec $M \in \Gamma^{\mathrm{EE'}}$ et $\forall t \in [0,T]$ solutions du sous-espace $\mathbf{\Gamma}$ regroupant :

- l'équilibre statique de l'interface soumise aux efforts exercés par les sous-domaines voisins : $\forall M \in \Gamma^{\mathrm{EE'}}$ et $\forall t \in [0,T]$

$$\mathbf{F}^{\mathrm{E}} + \mathbf{F}^{\mathrm{E'}} = \mathbf{0} \tag{1.16}$$

- la relation de comportement de l'interface : $\forall M \in \Gamma^{\mathrm{EE'}}$ et $\forall t \in [0,T]$

$$\mathcal{R}\left(\dot{\mathbf{W}}^{\mathrm{E}}, \dot{\mathbf{W}}^{\mathrm{E'}}, \mathbf{F}^{\mathrm{E}}, \mathbf{F}^{\mathrm{E'}}\right) = 0 \tag{1.17}$$

Il est possible de considérer plusieurs types de comportement pour les interfaces. Une brève description de certains d'entre eux est donnée dans la partie 3.1.5 page 50.

3.1.4 Algorithme LaTIn

En tenant compte de la décomposition en sous-domaines et interfaces ainsi que de la structuration du problème d'assemblage complet en « sous-problèmes », il est nécessaire pour résoudre le problème complet de mettre en œuvre un algorithme dédié (point **P2** de la méthode). Dans le cadre de la méthode LATIN, l'algorithme employé propose de chercher successivement les solutions associées aux deux sous-espaces $\mathbf{A_d}$ et $\mathbf{\Gamma}$ introduits précédemment. Pour les assemblages considérés dans ces travaux, cela revient à résoudre successivement les problèmes liés aux sous-domaines et aux interfaces.

Ces résolutions successives sont rendues viables grâce à l'emploi de directions de recherche notées E^+ et E^-. La figure 1.15 représente schématiquement le principe de l'algorithme itératif de la méthode LATIN. Le passage de la solution $\mathbf{s}_n$ à la solution $\mathbf{s}_{n+1}$ correspond à une itération de l'algorithme et à la résolution d'un problème dans les sous-domaines et d'un problème dans les interfaces. Ces deux résolutions sont désignées dans la méthode LATIN sous les termes :

- l'**étape locale** qui consiste à trouver $\widehat{\mathbf{s}}_n$ connaissant $\mathbf{s}_n$ et vérifiant : $\left|\begin{array}{r} \widehat{\mathbf{s}}_n \in \mathbf{\Gamma} \\ \widehat{\mathbf{s}}_n - \mathbf{s}_n \in \mathrm{E}^+ \end{array}\right.$
- l'**étape linéaire** (ou étape globale) qui consiste à trouver $\mathbf{s}_{n+1}$ connaissant $\widehat{\mathbf{s}}_n$ et vérifiant : $\left|\begin{array}{r} \mathbf{s}_{n+1} \in \mathbf{A_d} \\ \mathbf{s}_{n+1} - \widehat{\mathbf{s}}_n \in \mathrm{E}^- \end{array}\right.$

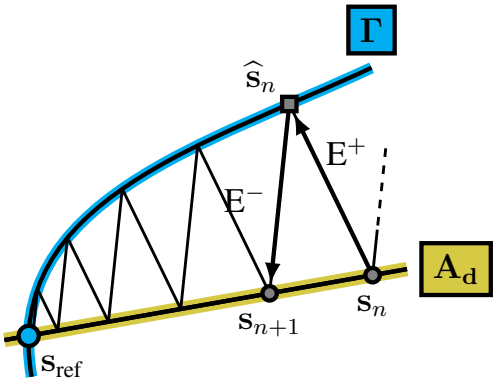


FIGURE 1.15 – Représentation schématique des itérations de l'algorithme LATIN

Remarque 1.6 : On distingue ici les grandeurs intervenant à l'étape locale de celles intervenant à l'étape linéaire : les grandeurs de l'étape locale (dans les interfaces) seront notées $\widehat{\bullet}$. Cette nouvelle notation fait apparaître les grandeurs $\widehat{\mathbf{W}}^{\mathrm{E}}$, $\mathbf{W}^{\mathrm{E}}$, $\widehat{\mathbf{F}}^{\mathrm{E}}$ et $\mathbf{F}^{\mathrm{E}}$. Ces grandeurs tendent grâce à l'algorithme à devenir égales : $\widehat{\mathbf{W}}^{\mathrm{E}} = \lim_{n\to+\infty} \mathbf{W}^{\mathrm{E}}$, $\widehat{\mathbf{F}}^{\mathrm{E}} = \lim_{n\to+\infty} \mathbf{F}^{\mathrm{E}}$ (et par conséquent $\widehat{\mathbf{s}}_n = \lim_{n\to+\infty} \mathbf{s}_n$)

■ **Directions de recherche :** les directions de recherche assurent le lien entre les inconnues solutions des sous-espaces $\mathbf{\Gamma}$ et $\mathbf{A_d}$. Dans le cadre du calcul d'assemblage, au niveau de l'étape locale, seules les grandeurs d'interfaces interviennent. Les directions de recherche portent donc uniquement sur ces quantités. Dans le cas où l'on considère un sous-domaine Ω_{E}, les directions de recherche sont conjuguées et définies telles que :

▶ à l'étape locale

$$\widehat{\mathbf{s}}_n - \mathbf{s}_n \in \mathrm{E}^+ \Leftrightarrow \widehat{\mathbf{F}}^{\mathrm{E}} - \mathbf{F}^{\mathrm{E}} = \mathbf{k}^+ \left(\widehat{\dot{\mathbf{W}}}^{\mathrm{E}} - \dot{\mathbf{W}}^{\mathrm{E}} \right) \tag{1.18}$$

▶ à l'étape linéaire

$$\mathbf{s}_{n+1} - \widehat{\mathbf{s}}_n \in \mathrm{E}^- \Leftrightarrow \mathbf{F}^{\mathrm{E}} - \widehat{\mathbf{F}}^{\mathrm{E}} = -\mathbf{k}^- \left(\dot{\mathbf{W}}^{\mathrm{E}} - \widehat{\dot{\mathbf{W}}}^{\mathrm{E}} \right) \tag{1.19}$$

où $\mathbf{k}^+$ et $\mathbf{k}^-$ sont deux opérateurs symétriques définis positifs qui sont des paramètres de la méthode. Dans le cas des calculs d'assemblages, ils sont classiquement choisis tels que $\mathbf{k}^+ = \mathbf{k}^- = k_0\mathbf{I}$ où k est un scalaire. Une interprétation mécanique de ces paramètres peut être trouvée dans [CHAMPANEY 1996].

■ **Convergence :** la convergence de la stratégie monoéchelle a été démontrée dans [LADEVÈZE 1999] pour le type de problèmes traités ici. Bien que la valeur des paramètres des directions de recherche n'ait aucun impact sur la solution du problème, elle a un impact extrêmement important sur le taux de convergence de l'algorithme. Elle doit donc être choisie judicieusement. Dans le cadre des problèmes quasi-statiques considérés ici, on choisit k_0 tel que

$$k_0 = \frac{ET}{L_c} \tag{1.20}$$

où E est le module d'Young, $[0, T]$ l'intervalle de temps considéré et L_c la plus grande dimension du sous-domaine Ω_{E}.

■ **Critère d'arrêt :** afin de contrôler la convergence de l'algorithme et de pouvoir la stopper, un indicateur d'erreur est employé. Ce dernier est une mesure énergétique entre $\mathbf{s}$ et $\widehat{\mathbf{s}}$ et quantifie la non-satisfaction des relations de comportement des interfaces pour une solution approchée $\mathbf{s}^n$ de l'étape globale qui vérifie déjà toutes les autres équations. Dans le cas de l'élasticité, un indicateur simple d'erreur en énergie est construit à partir des champs surfaciques. L'indicateur d'erreur global s'écrit donc en exploitant la norme suivante définie sur les interfaces [CHAMPANEY 1996] :

$$\eta = \frac{\sum_E \|\mathbf{s}^E - \widehat{\mathbf{s}}^E\|^2}{\sum_E \|\mathbf{s}^E\| + \|\widehat{\mathbf{s}}^E\|^2}$$

$$\text{avec } \|\mathbf{s}\|^2 = \int_0^T \int_{\partial\Omega_{\mathrm{E}}} \mathbf{F}^{\mathrm{E}T} k_0^{-1} \mathbf{F}^{\mathrm{E}} dS dt + \int_0^T \int_{\partial\Omega_{\mathrm{E}}} \mathbf{W}^{\mathrm{E}T} k_0 \mathbf{W}^{\mathrm{E}} dS dt \tag{1.21}$$

Cet indicateur tend vers 0 lorsque l'algorithme converge. En effet, s^E et $\widehat{\mathrm{s}}^E$ (soient $(\mathbf{F}^{\mathrm{E}}, \mathbf{W}^{\mathrm{E}})$ et $(\widehat{\mathbf{F}}^{\mathrm{E}}, \widehat{\mathbf{W}}^{\mathrm{E}})$) sont confondues à convergence (plus précisément, lorsque n tend vers l'infini).

3.1.5 Lois de comportement des interfaces

La décomposition en interfaces et sous-domaines retenue pour l'approche LATIN rend la gestion des comportements des interfaces plus aisée. En effet, la stratégie permet de gérer les interfaces indépendamment des sous-domaines tout en assurant la communication des grandeurs entre sous-domaines et interfaces. Les différents comportements d'interfaces sont brièvement rappelés ici. On pourra se référer à [BLANZÉ, CHAMPANEY, COGNARD et al. 1996] pour des détails complémentaires.

Afin de mettre en place les relations de comportement aux interfaces, nous considérerons les notations suivantes :

- $\forall M \in \Gamma^{\mathrm{EE'}},\ \mathbf{N} = \mathbf{N}(M)$ la normale en tout point de l'interface $\Gamma^{\mathrm{EE'}}$;
- $\forall M \in \Gamma^{\mathrm{EE'}},\ \mathbf{\Pi} = \mathbf{\Pi}(M)$ l'opérateur de projection dans le plan tangent en tout point de l'interface $\Gamma^{\mathrm{EE'}}$ défini tel que, soit $\mathbf{w} = \mathbf{w}(M)$ un vecteur $\mathbf{\Pi w} = \mathbf{w} - (\mathbf{N}^T\mathbf{w})\mathbf{N}$;
- $\forall M \in \Gamma^{\mathrm{EE'}}, \forall t \in [0,T],\ \left[\!\left[\widehat{\mathbf{W}}\right]\!\right]_{\mathrm{EE'}} = \widehat{\mathbf{W}}^{\mathrm{E'}} - \widehat{\mathbf{W}}^{\mathrm{E}}$ et $\left[\!\left[\dot{\widehat{\mathbf{W}}}\right]\!\right]_{\mathrm{EE'}} = \dot{\widehat{\mathbf{W}}}^{\mathrm{E'}} - \dot{\widehat{\mathbf{W}}}^{\mathrm{E}}$ respectivement les sauts de déplacement et de vitesse à l'interface $\Gamma^{\mathrm{EE'}}$.

■ **Interface « parfaite » :** on considère une interface comme « parfaite » si le saut de déplacement au travers de l'interface est nul, c'est-à-dire si

$$\forall M \in \Gamma^{\mathrm{EE'}}, \forall t \in [0,T],\ \left[\!\left[\widehat{\mathbf{W}}\right]\!\right]_{\mathrm{EE'}} = \mathbf{0} \tag{1.22}$$

■ **Interface avec contact :** la présence de contact au niveau d'une interface est modélisée à l'aide d'une relation de comportement d'interface basée sur deux conditions de contact s'écrivant :

- **Décollement :** si $\mathbf{N}^T \left[\!\left[\widehat{\mathbf{W}}\right]\!\right]_{\mathrm{EE'}} > 0$ alors $\mathbf{N}^T\widehat{\mathbf{F}}^{\mathrm{E}} = \mathbf{N}^T\widehat{\mathbf{F}}^{\mathrm{E'}} = 0$;
- **Contact :** si $\mathbf{N}^T \left[\!\left[\widehat{\mathbf{W}}\right]\!\right]_{\mathrm{EE'}} = 0$ alors $\mathbf{N}^T\widehat{\mathbf{F}}^{\mathrm{E}} + \mathbf{N}^T\widehat{\mathbf{F}}^{\mathrm{E'}} = 0$.

De manière pratique, la vérification des conditions de contact peut être écrite en vérifiant le signe de la fonction :

$$C_N = \frac{1}{2}\mathbf{N}^T \left[\!\left[\widehat{\mathbf{W}}\right]\!\right]_{\mathrm{EE'}} - \frac{1}{2k_0}\mathbf{N}^T(\widehat{\mathbf{F}}^{\mathrm{E'}} - \widehat{\mathbf{F}}^{\mathrm{E}}) \tag{1.23}$$

Les conditions de contact se traduisent alors de la manière suivante :

- **Décollement :** si $C_N > 0$;
- **Contact :** si $C_N \leq 0$.

Remarque 1.7 : Il est possible d'étendre le comportement des interfaces avec contact en tenant compte d'un jeu initial. Afin de ne pas surcharger les rappels introduits ici, cette extension n'est pas développée mais peut être retrouvée dans [CHAMPANEY 1996].

■ **Interface avec frottement :** la prise en compte du frottement au niveau d'une interface peut être réalisée en vérifiant les conditions de frottement de Coulomb s'écrivant :

► **Adhérence :** si $\|\boldsymbol{\Pi}\widehat{\mathbf{F}}^{\mathrm{E}}\| < \mu|\mathbf{N}^T\widehat{\mathbf{F}}^{\mathrm{E}}|$ alors $\boldsymbol{\Pi}\left[\!\left[\dot{\widehat{\mathbf{W}}}\right]\!\right]_{\mathrm{EE'}} = 0$;

► **Glissement :** si $\|\boldsymbol{\Pi}\widehat{\mathbf{F}}^{\mathrm{E}}\| = \mu|\mathbf{N}^T\widehat{\mathbf{F}}^{\mathrm{E}}|$ alors $\exists\lambda > 0,\ \boldsymbol{\Pi}\left[\!\left[\dot{\widehat{\mathbf{W}}}\right]\!\right]_{\mathrm{EE'}} = -\lambda\widehat{\mathbf{F}}^{\mathrm{E}}$.

De manière pratique, la vérification des conditions de contact peut être écrite en vérifiant le signe de la fonction :

$$\mathbf{G}_T = \frac{k_0}{2}\boldsymbol{\Pi}\left[\!\left[\dot{\widehat{\mathbf{W}}}\right]\!\right]_{\mathrm{EE'}} - \frac{1}{2}\boldsymbol{\Pi}(\widehat{\mathbf{F}}^{\mathrm{E'}} - \widehat{\mathbf{F}}^{\mathrm{E}}) \tag{1.24}$$

Les conditions de frottement se traduisent alors de la manière suivante :

► **Adhérence :** si $\|\mathbf{G}_T\| \leq \mu|\mathbf{N}^T\widehat{\mathbf{F}}^{\mathrm{E}}|$

► **Glissement :** si $\|\mathbf{G}_T\| > |\mathbf{N}^T\widehat{\mathbf{F}}^{\mathrm{E}}|$

Remarque 1.8 : La prise en compte du frottement se fait dans le cas de la méthode LATIN sans les difficultés rencontrées dans les approches classiques (la direction du glissement par exemple est obtenue via l'étape globale).

3.1.6 Aspects pratiques de la résolution et spécificités de la méthode

Quelques aspects pratiques de la méthode employée sont rappelés ici :

■ **Discrétisation :** la méthode LATIN pour le calcul d'assemblages complexes utilise une discrétisation « éléments finis » standard. Au niveau des interfaces, une discrétisation compatible est utilisée. Elle nécessite une attention particulière lors du maillage des sous-domaines afin de s'assurer de la coïncidence des noeuds du maillage composant les bords des sous-domaines en vis-à-vis au niveau d'une interface.

■ **Résolution de l'étape locale :** la présence d'interfaces non parfaites (avec frottement et/ou contact) nécessite une attention particulière. Bien que la construction des indicateurs C_N (eq. 1.23) et $\mathbf{G}_T$ (eq. 1.24) simplifie grandement la prise en compte de ces phénomènes non-linéaires et permet de connaître le statut (décollement ou contact, adhérence ou glissement) de chaque point de l'interface, quelques manipulations peuvent être requises. En particulier, l'indicateur de contact C_N nécessite l'intégration (ici implicite) du champ de vitesse afin d'obtenir le champ de déplacement à l'interface.

■ **Résolution de l'étape globale :** la prise en compte de la discrétisation au niveau de l'étape globale (c'est-à-dire dans les sous-domaines) ne pose pas de difficulté particulière. Il est néanmoins important de constater que les opérateurs (matrice de rigidité et le terme de bord relatif aux interfaces) qui interviennent à cette étape restent constants. Cette information est capitale dans le fonctionnement de la stratégie multiparamétrique qui sera introduite dans la partie 3.2.3 page 59.

■ **Logiciel employé :** cette approche LATIN a fait l'objet de nombreux développements sous la forme de logiciels et de démonstrateurs basés pour certains sur les librairies développées au LMT-Cachan. Dans le cadre de ces travaux, le logiciel utilisé est COFAST3D développé par Laurent Champaney [CHAMPANEY 1996]. Ce logiciel se base sur les routines proposées dans le logiciel Cast3m [6] développé par le CEA et intègre des procédures en langage Gibiane. Elles permettent de réaliser facilement la mise en données d'un problème d'assemblage en permettant une définition aisée des sous-domaines et interfaces ainsi que la détection automatique des ces dernières. Les procédures permettent également la résolution du problème d'assemblage à l'aide de l'algorithme LATIN, le post-traitement des résultats obtenus et l'utilisation de la stratégie multiparamétrique introduite dans la partie 3.2.3 page 59.

3.1.7 Quelques compléments sur la méthode LATIN

Aux différents éléments rappelés dans les parties précédentes, la méthode LATIN telle qu'elle fut introduite peut se voir complétée d'une échelle macroscopique [LADEVÈZE, LOISEAU et DUREISSEIX 2001]. Celle-ci permet d'assurer une réelle extensibilité numérique [7] à l'approche mixte proposée dans la méthode. La méthode multi-échelle repose sur une décomposition des grandeurs d'interfaces en une composante microscopique et une composante macroscopique. Un choix pertinent des quantités macroscopiques assure l'accélération de la convergence de la méthode [VIOLEAU 2007]. Par ailleurs, il est possible d'ajouter une représentation des inconnues (point **P3** de la méthode LATIN). Connue sous les noms d'approximation radiale [LADEVÈZE 1985] ou d'approximation PGD (*Proper Generalized Decomposition*), cette technique consiste à décrire les champs inconnus comme une somme de produits de fonctions de l'espace et du temps (voir par exemple [PASSIEUX 2008]).

Ces deux stratégies ne seront pas employées dans le cadre de ces travaux de thèse.

6. `www-cast3m.cea.fr`

7. Une méthode de décomposition de domaine est dite extensible numériquement lorsque son taux de convergence est inchangé avec l'augmentation du nombre de sous-domaines tout en maintenant le rapport $\frac{h}{H}$ constant (h et H désignent respectivement une longueur caractéristique du sous-domaine et une longueur caractéristique du domaine complet).

3.2 Une méthode de calcul pour les études paramétriques ou l'optimisation

Dans le contexte des études paramétriques ou dans celui de problèmes d'optimisation, il est nécessaire d'évaluer la réponse d'un modèle pour plusieurs jeux de paramètres. Dans les travaux présentés dans ce manuscrit, on s'intéresse à des paramètres tels que des coefficients de frottement, des jeux et des précharges. Les problèmes résultant de la prise en compte des variabilités de ce type de paramètres sont très similaires : ils n'engendrent pas de modifications de la géométrie et leur action est très locale. Cependant, ils peuvent avoir une influence à l'échelle globale de la structure. Par exemple, la modification des caractéristiques de frottement au niveau d'une liaison boulonnée peut engendrer une modification notoire et globale du comportement de celle-ci : les efforts transmis par adhérence de la liaison pourraient être transmis par l'intermédiaire d'une ou plusieurs vis travaillant en cisaillement.

L'obtention d'une réponse du modèle peut nécessiter un temps de calcul extrêmement important du fait :

- de la taille du problème qui, si on le souhaite représentatif d'un assemblage complexe, peut comporter un nombre important d'interfaces, un nombre important de degrés de liberté aux interfaces et dans les pièces.
- de la nécessité de prendre en compte les comportements non-linéaires pouvant survenir aux interfaces.
- de la prise en compte de chargements ayant des évolutions complexes qui nécessitent une discrétisation temporelle conséquente.
- de la nécessité de s'assurer d'une convergence suffisante de la méthode itérative employée (afin de minimiser les erreurs locales commises).

En partant de ce constat, il n'est *a priori* pas viable de recalculer l'intégralité de la solution du problème d'assemblage considéré pour chaque jeu de paramètres. Cependant, il est possible de tenir compte des spécificités d'un calcul réalisé par la Méthode des Éléments Finis comme par exemple la constance de certains opérateurs ou au contraire leur dépendance connue aux paramètres. Les méthodes brièvement rappelées dans la suite se démarquent par le type de paramètres qu'elles peuvent prendre en compte en proposant une modélisation appropriée. On distinguera les méthodes adaptées aux variables aléatoires et celles adaptées aux variables déterministes.

Par ailleurs, aux études paramétriques et à l'optimisation peuvent s'ajouter les analyses de sensibilité (qui visent à évaluer la variabilité de la réponse du modèle ou du système considéré en fonction de paramètres déterministes ou aléatoires) et les analyses de fiabilité (qui visent à évaluer la probabilité des défaillances d'un système en tenant compte de performances à atteindre).

Dans cette partie, plusieurs méthodes dédiées à cette problématique de multiples résolutions seront rappelées. Une attention particulière sera portée sur la stratégie multi-

paramétrique qui est capable de tenir compte des spécificités de la méthode LATIN afin de réduire les coûts de calcul.

3.2.1 Quelques stratégies pour les multiples résolutions

■ **Méthodes fréquentielles :** Les méthodes fréquentielles apparaissent ici comme les plus basiques. En considérant, $\mathbf{x} = (x_1, x_2, \ldots, x_{n_p})$ les n_p paramètres du problème, on peut envisager deux approches :

- une **approche énumérative** où le domaine de variation de chaque paramètre x_i est subdivisé en n_i valeurs déterministes. L'étude complète nécessitera alors N calculs (avec $N = \prod_i^{n_p} n_i$).
- une **approche probabiliste** de type
  - Monte-Carlo consistant à tirer aléatoirement des valeurs de chaque paramètre x_i. Ce type d'approche nécessite une connaissance *a priori* des lois de répartition statistiques des paramètres et un nombre important de tirages pour converger.
  - quasi-Monte-Carlo permettant d'accélérer la convergence des approches Monte Carlo pour des problèmes comportant peu de paramètres en utilisant des tirages pseudo-aléatoires.

  Le principal défaut de ces méthodes fréquentielles est qu'elles nécessitent un nombre de résolutions qui reste important. Elles ont cependant l'avantage de ne pas être intrusives et, dans le cas des approches probabilistes de présenter une grande robustesse dans des cas aussi bien linéaires que non-linéaires.

■ **Méthodes de perturbation :** Les méthodes de perturbation consistent à considérer de faibles variations des paramètres autour de valeurs particulières de ces derniers. Dans un contexte où les paramètres sont aléatoires, les variations de ces derniers sont souvent considérées autour de leur valeurs moyennes. Les paramètres considérés et les opérateurs de la méthode éléments finis qui en découlent sont alors décomposés en utilisant des séries de Taylor ou de Neumann. Le comportement de la structure et de la réponse étudiées est alors analysé en termes de variance et de moyenne.

Afin d'expliciter ce principe, on considérera un problème élément finis « simple » réduit à la forme

$$\mathbf{KU} = \mathbf{F} \tag{1.25}$$

où $\mathbf{K}$, $\mathbf{U}$ et $\mathbf{F}$ sont respectivement la matrice de rigidité globale, le vecteur des déplacement nodaux et celui des efforts nodaux. On supposera que $\mathbf{K}$ et $\mathbf{U}$ sont dépendants des paramètres $\mathbf{x}$. Il est alors possible d'employer :

- une méthode basée sur les **séries de Taylor** [HISADA et NAKAGIRI 1985 ; SHINOZUKA 1987] où les paramètres sont écrits tels que $\forall i \in [\![0, n_p]\!],\ x_i = \overline{x_i} + \alpha_i$ avec

$\mathbb{E}[x_i] = \overline{x_i}$ et $\text{Var}[x_i] = \alpha_i$ [8]. En supposant que le problème considéré est linéaire et que les variations autour de la moyenne $\overline{x_i}$ sont petites, il est alors possible de décomposer $\mathbf{K}$ et $\mathbf{U}$ en séries de Taylor autour des $\overline{x_i}$:

$$\mathbf{K}(\mathbf{x}) = \mathbf{K}^0 + \sum_{k=1}^{n_p} \mathbf{K}_k^I \alpha_k + \frac{1}{2} \sum_{k=1}^{n_p} \sum_{l=1}^{n_p} \mathbf{K}_{kl}^{II} \alpha_k \alpha_l + o\left(\|\boldsymbol{\alpha}\|^2\right) \tag{1.26}$$

$$\mathbf{U}(\mathbf{x}) = \mathbf{U}^0 + \sum_{k=1}^{n_p} \mathbf{U}_k^I \alpha_k + \frac{1}{2} \sum_{k=1}^{n_p} \sum_{l=1}^{n_p} \mathbf{U}_{kl}^{II} \alpha_k \alpha_l + o\left(\|\boldsymbol{\alpha}\|^2\right) \tag{1.27}$$

avec

$$\begin{array}{llll} \mathbf{K}^0 = [\mathbf{K}]_{\boldsymbol{\alpha}=\mathbf{0}} & \mathbf{K}_k^I = \left[\dfrac{\partial \mathbf{K}}{\partial \alpha_k}\right]_{\boldsymbol{\alpha}=\mathbf{0}} & \mathbf{K}_{kl}^{II} = \left[\dfrac{\partial^2 \mathbf{K}}{\partial \alpha_k \partial \alpha_l}\right]_{\boldsymbol{\alpha}=\mathbf{0}} \\ \mathbf{U}^0 = [\mathbf{U}]_{\boldsymbol{\alpha}=\mathbf{0}} & \mathbf{U}_k^I = \left[\dfrac{\partial \mathbf{U}}{\partial \alpha_k}\right]_{\boldsymbol{\alpha}=\mathbf{0}} & \mathbf{U}_{kl}^{II} = \left[\dfrac{\partial^2 \mathbf{U}}{\partial \alpha_k \partial \alpha_l}\right]_{\boldsymbol{\alpha}=\mathbf{0}} \end{array} \tag{1.28}$$

L'injection des décompositions données par les équations 1.26 et 1.27 dans l'équation 1.25 et une identification terme à terme conduisent à l'écriture du système d'équations suivant :

$$\begin{cases} \mathbf{K}_0 \mathbf{U}_0 = \mathbf{F} \\ \mathbf{K}_0 \mathbf{U}_k^I = -\mathbf{K}_k^I \mathbf{U}^0 \\ \mathbf{K}^0 \mathbf{U}_{kl}^{II} = -\left(\mathbf{K}_k^I \mathbf{U}_l^I + \mathbf{K}_l^I \mathbf{U}_k^I + \mathbf{K}_{kl}^{II} \mathbf{U}^0\right) \\ \qquad \vdots \end{cases} \tag{1.29}$$

Remarque 1.9 : On suppose ici que le second membre $\mathbf{F}$ est déterministe et donc ne nécessite pas de décomposition.

L'avantage de cette approche réside dans le fait que la matrice $\mathbf{K}^0$ n'est calculée qu'une fois puis réutilisée dans toutes les équations. De cette décomposition, il est possible de récupérer facilement des informations statistiques portant sur les grandeurs issues de la résolution (telles que les déplacements, contraintes et déformations). L'inconvénient majeur de ce type de méthodes est la nécessité de calculer les gradients qui peuvent engendrer des problèmes de convergence [F. YAMAZAKI et al. 1988] en particulier dans le cas de problèmes non-linéaires [HISADA 1988]. La restriction aux petites variations et l'apparition d'instabilités numériques lorsque l'ordre de la décomposition augmente sont également des éléments limitants de cette méthode.

Par ailleurs, les adaptations de cette méthode fréquentielle aux problèmes non-linéaires n'ont été réalisées que dans des cas très spécifiques [DEODATIS 1989].

8. $\mathbb{E}[\bullet]$ et $\text{Var}[\bullet]$ sont respectivement l'espérance et la variance statistiques.

- une méthode basée sur le **développement de Neumann** limitée au cas linéaire [SHINOZUKA 1987 ; F. YAMAZAKI et al. 1988] où la matrice de rigidité globale $\mathbf{K}(\mathbf{x})$ est décomposée en une partie déterministe $\mathbf{K}_0$ et une partie aléatoire $\Delta\mathbf{K}$:

$$\mathbf{K} = \mathbf{K}_0 + \Delta\mathbf{K} = \mathbf{K}_0(\mathbf{I} + \mathbf{P}) \text{ où } \mathbf{P} = \mathbf{K}_0^{-1}\Delta\mathbf{K} \tag{1.30}$$

La technique consiste alors à écrire l'inverse de la matrice de rigidité $\mathbf{K}^{-1} = \left(\mathbf{K}^0 + \Delta\mathbf{K}\right)^{-1} = (\mathbf{I} + \mathbf{P})^{-1}\mathbf{K}_0^{-1}$ puis à décomposer l'opérateur $(\mathbf{I} + \mathbf{P})$ en série de Neumann :

$$(\mathbf{I} + \mathbf{P})^{-1} = \sum_{k=0}^{\infty}(-1)^k\mathbf{P}^k \tag{1.31}$$

La résolution du système linéaire associé au problème éléments finis s'écrit alors :

$$\mathbf{U}(\mathbf{x}) = \mathbf{K}(\mathbf{x})^{-1}\mathbf{F} = \sum_{k=0}^{\infty}(-1)^k\mathbf{P}^k\mathbf{K}_0^{-1}\mathbf{F} \tag{1.32}$$

Ainsi, en tronquant la décomposition à l'ordre m, il est possible d'obtenir itérativement les termes d'une suite géométrique de raison $(-\mathbf{P})$ et de premier terme $\mathbf{K}_0^{-1}\mathbf{F}$. La détermination de la solution s'obtient donc de la manière suivante :

$$\mathbf{U}(\mathbf{x}) \approx \sum_{k=0}^{m}\mathbf{U}_k \text{ où } \begin{cases} \mathbf{U}_0 = \mathbf{K}_0^{-1}\mathbf{F} \\ \mathbf{U}_k = -\mathbf{P}\mathbf{U}_{k-1} \text{ avec } k > 0 \end{cases} \tag{1.33}$$

A l'instar de la méthode basée sur le développement en série de Taylor, cette approche permet, dans le cas où l'on se place dans une démarche de Monte-Carlo, de n'avoir à construire qu'une seule fois la matrice $\mathbf{K}_0$ pour l'ensemble des échantillons tirés. En effet, dans ce cas, seule la valeur de la matrice $\Delta\mathbf{K}$ est impactée par les différentes valeurs des paramètres tirés.

■ **Méthodes spectrales :** Les méthodes spectrales dites stochastiques [GHANEM et SPANOS 1991] permettent la propagation d'incertitudes sur les paramètres au sein d'un modèle mathématique. De ces méthodes ont découlé les méthodes de type élément finis stochastiques spectraux. Elles s'appuient sur la **décomposition de Karhunen-Loève** et les fonctions de **chaos polynomial**.

La décomposition de Karhunen-Loève [LOÉVE 1977] est employée pour approximer les champs aléatoires connus (par exemple des coefficients matériaux...). Ces derniers sont alors approchés sur les modes propres du noyau de leur matrice de covariance. Classiquement, en considérant un paramètre matériau local aléatoire dont la fonction de densité de probabilité est connue, il est possible de formuler, dans le cadre de l'approche éléments finis proposée, la matrice de raideur globale sous la forme suivante (dans le cas où le développement est tronqué à l'ordre m) :

$$\mathbf{K}(\theta) = \mathbf{K}_0 + \sum_{i=1}^{m}\xi_i(\theta)\mathbf{K}_i \tag{1.34}$$

où $\mathbf{K}_0$ et les $\mathbf{K}_i$ sont respectivement la matrice de rigidité moyenne et des matrices déterministes résultant de la méthode de décomposition de Karhunen-Loève.

Si le processus stochastique n'est pas connu, ce qui est typiquement le cas pour les déplacements, le champ est recherché par projection sur la base du chaos polynomial à l'ordre p $\{\psi_k\}_{k=0}^{p}$ [WIENER 1938]. Il est alors possible de rechercher les déplacements nodaux aléatoires éléments finis sous la forme :

$$\mathbf{U}(\theta) = \sum_{i=0}^{p} \mathbf{U}_i \psi_i(\theta) \tag{1.35}$$

où les $\mathbf{U}_i$ sont les déplacements nodaux et $\text{cov}[\psi_i(\theta), \psi_j(\theta)] = c_i \delta_{ij}$ [9].

En supposant le chargement déterministe et les décompositions introduites ci-dessus, le problème à résoudre devient :

$$\begin{gathered} \left(\mathbf{K}_0 + \sum_{i=1}^{M} \xi_i(\theta)\mathbf{K}_i\right)\left(\sum_{i=0}^{p} \mathbf{U}_i \psi_k(\theta)\right) = \mathbf{F} \\ \text{soit} \sum_{i=1}^{M}\sum_{j=0}^{p} \mathbf{K}_i \mathbf{U}_j \xi_i(\theta)\psi_j(\theta) = \mathbf{F} \end{gathered} \tag{1.36}$$

Finalement, l'emploi de la propriété d'orthogonalité des fonctions du chaos polynomial conduit au système linéaire de grande taille ($p \times$ nombre de degrés de liberté) présenté sous forme matricielle par l'équation 1.37.

$$\begin{bmatrix} \mathbf{K}^{(00)} & \mathbf{K}^{(01)} & \dots & \mathbf{K}^{(0p)} \\ \dots & \mathbf{K}^{(11)} & \dots & \mathbf{K}(1p) \\ \dots & \vdots & \ddots & \vdots \\ \mathbf{K}^{(p0)} & \mathbf{K}^{(p1)} & \dots & \mathbf{K}^{(pp)} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_p \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{1.37}$$

avec $\mathbf{K}^{(kl)} = \sum_{i=0}^{M} \mathbb{E}[\xi_i \psi_k \psi_l]\mathbf{K}_i$

■ **Réduction de modèles :** De plus en plus employées actuellement, les méthodes de réduction de modèle sont conçues dans le but de réduire les temps de calcul en proposant la construction de bases de fonctions de l'espace, du temps et éventuellement des paramètres dont la combinaison linéaire permettent d'approcher la solution. Parmi les méthodes classiquement employées, on s'intéresse particulièrement à la POD (*Proper Orthogonal Decomposition*) [LUMLEY 1967] et à la PGD (*Proper Generalized Decomposition*) [LADEVÈZE 1985 ; BEYLKIN et MOHLENKAMP 2005 ; AMMAR et al. 2006 ; AMMAR et al. 2007].

La POD cherche à construire une base réduite permettant d'approcher la solution du problème tout en minimisant l'écart à cette solution. Elle s'appuie classiquement

9. $\text{cov}[\bullet, \bullet']$ désigne la fonction de covariance statistique.

sur la construction d'une matrice obtenue en un ensemble de *snapshots* (en certains instants du temps par exemple). Une décomposition SVD fournit alors les coefficients et la base de la décomposition [CHATTERJEE 2000]. Il est également possible de ne choisir d'employer qu'une base réduite. La ROM-POD (*Reduced Order Model - Proper Orthogonal Decomposition*) [MADAY, PATERA et TURINICI 2002] permet ensuite de construire une approximation de la solution sur l'ensemble de l'espace et du temps. L'évaluation de la fonction approchée est alors extrêmement rapide mais elle a nécessité la construction de la base (qui est coûteuse en temps).

La PGD propose de construire ces bases au fur et à mesure du calcul [CHINESTA, LADEVÈZE et CUETO 2011]. La nécessité d'ajouter un nouveau « mode » base est contrôlée par la méthode directement. Des extensions de la PGD proposant une décomposition sur une base de fonctions des paramètres de l'étude considérée ont vu le jour ces dernières années [AMMAR, CHINESTA et al. 2012]. Par ailleurs, des couplages entre d'un côté la méthode LATIN et la PGD [LADEVÈZE 1985] et de l'autre entre la stratégie multiparamétrique et la PGD [HEYBERGER, BOUCARD et NÉRON 2012 ; HEYBERGER, BOUCARD et NÉRON 2013] ont permis une réduction importante des coûts de calcul pour la construction des différentes fonctions de l'espace et du temps.

La PGD adaptée aux décompositions paramétriques consiste à construire une base de fonctions des paramètres pour pouvoir ensuite approcher à moindre coût la réponse du système pour un jeu de paramètres quelconque. En considérant une solution $\mathbf{u}$ d'un problème fonction de n_p paramètres x_i, la décomposition prend la forme

$$\mathbf{u}(\mathbf{x}) \approx \sum_{i=1}^{N} X_i^1(x_1) \cdot X_i^2(x_2) \cdots X_i^{n_p}(x_{n_p}) \tag{1.38}$$

On pourra se référer à [CHINESTA, LEYGUE et al. 2013] concernant l'emploi de la PGD dans un cadre de décomposition paramétrique dédiée.

3.2.2 Approches numériques pour les multiples résolutions

Bien que performantes, les méthodes proposées précédemment ne s'appliquent qu'à des cas spécifiques tels que pour les approches stochastiques ou pour de faibles variations des paramètres. Par ailleurs, l'application de ces méthodes à des cas fortement non-linéaires et leur forte intrusivité les rendent difficilement exploitables dans le cadre de l'optimisation.

Cependant, dans le cas de problèmes conduisant à la formulation de systèmes linéaires, il est possible d'avoir une vision plus numérique du problème en cherchant à tirer partie des spécificités mises en œuvre pour résoudre ce système linéaire.

■ **Solveurs de Krylov :** Dans de nombreux cas, la résolution du système linéaire $\mathbf{A}\mathbf{x} = \mathbf{b}$ est réalisée par l'intermédiaire de solveurs de Krylov [SAAD 2000] tels que l'algorithme du Gradient Conjugué ou l'algorithme GMRES (*Generalized Minimal RESidual*). Ces algorithmes reposent sur la génération d'une base d'un sous-espace de Krylov de $\mathbf{A}$. Dans le cas où on l'on doit résoudre une série de systèmes linéaires $\mathbf{A}_i\mathbf{x} = \mathbf{b}_i$ proches les uns des autres, il est possible d'accélérer la convergence des algorithmes en réutilisant

les sous-espaces des résolutions précédentes. Un certain nombre de méthodes d'accélération de ce type peuvent être utilisées pour améliorer l'initialisation des algorithmes, le préconditionneur et l'espace d'augmentation [RISLER et REY 2000 ; SAAD 2000]. La méthode SRKS (*Selective Reuse of Krylov Subspaces*) en particulier propose une extraction des grandeurs représentatives du comportement spectral du système permettant ensuite d'accélérer les résolutions suivantes [CHAPMAN et SAAD 1997].

■ **SLDLT :** La méthode SLDLT [NOTIN et al. 2011] propose une accélération des résolutions lorsque plusieurs systèmes linéaires sont à résoudre. Pour cela, elle se base sur l'algorithme de résolution de systèmes linéaires LDL. Ce dernier propose une décomposition de la matrice symétrique définie positive $\mathbf{A}$ en une matrice triangulaire inférieure $\mathbf{L}$ et une matrice diagonale $\mathbf{D}$ telle que

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^T \tag{1.39}$$

La solution du système $\mathbf{A}\mathbf{x} = \mathbf{b}$ est alors cherchée de la manière suivante

$$\mathbf{x} = \mathbf{L}^{-T}\mathbf{D}^{-1}\mathbf{L}^{-1}\mathbf{b} \tag{1.40}$$

Lorsque l'on cherche à résoudre un ensemble de problèmes proches $\mathbf{A}_i\mathbf{x} = \mathbf{b}_i$, la méthode SLDLT propose d'approcher la matrice $\mathbf{A}_i$ de la manière suivante :

$$\mathbf{A}_i \approx \tilde{\mathbf{A}}_i = \mathbf{L}_0\overline{\mathbf{D}}_i\mathbf{L}_0^T \tag{1.41}$$

Cette approche permet d'éviter la détermination coûteuse des matrices $\mathbf{L}_i$ associées à chaque matrice $\mathbf{A}_i$. Le problème consiste alors à chercher la matrice $\overline{\mathbf{D}}_i$ par l'intermédiaire d'un problème d'optimisation visant à minimiser l'erreur entre les matrices $\tilde{\mathbf{A}}_i$ et $\mathbf{A}_i$.

3.2.3 La stratégie multiparamétrique

Comme nous l'avons vu précédemment, la méthode LATIN se détache des méthodes de résolution éléments finis classiques par l'emploi d'un algorithme de résolution particulier. La stratégie employée et la décomposition en sous-espaces permettent d'obtenir à chaque itération de l'algorithme une solution du problème sur l'ensemble du temps et de l'espace. De plus, la solution est entièrement définie par les champs de bord des sous-domaines. Enfin, la méthode repose sur un ensemble d'opérateurs qui restent constants dans le cadre de ces travaux où on ne considère que des paramètres variables au niveau des interfaces.

La stratégie multiparamétrique [BOUCARD et LADEVÈZE 1999] présentée ici a été introduite et employée principalement dans un contexte de structures assemblées comportant des problèmes de contact et frottement multiples entre pièces. Elle a été initialement introduite pour construire des surfaces de réponse [BOUCARD 2001], pour réaliser des études paramétriques et des analyses de sensibilité [BOUCARD et CHAMPANEY 2003 ; BOUCARD et CHAMPANEY 2004 ; CHAMPANEY, BOUCARD et GUINARD 2008], puis dans un contexte de prise en compte d'incertitudes dans la modélisation des assemblages [BLANZÉ, CHAMPANEY et BOUCARD 2004 ; CHAMPANEY, BLANZÉ et BOUCARD 2005],

de variations de chargement et de détails géométriques [CHAMPANEY et BOUCARD 2007], de présence-absence de pièces d'un assemblage [CHAMPANEY 2012] et pour l'optimisation paramétrique avec calculs convergés [BUYTET, BOUCARD et GUIDAULT 2008 ; BOUCARD, BUYTET et GUIDAULT 2009 ; CHAMPANEY 2011 ; SOULIER et BOUCARD 2013] et partiellement convergés [SOULIER et BOUCARD 2012 ; COURRIER, BOUCARD et SOULIER 2013]. Elle fut également employée dans le contexte d'études paramétriques d'assemblages de structures composites endommageables [ROULET, BOUCARD et CHAMPANEY 2013] et de pièces ayant un comportement élasto-viscoplastique [RELUN, NÉRON et BOUCARD 2012]. Elle a enfin été mise en œuvre pour la simulation d'assemblages en dynamique transitoire [ODIÉVRE 2009 ; BOUCARD, ODIÉVRE et GATUINGT 2011]. Plus récemment, elle a été mise en œuvre pour réaliser des études paramétriques en employant la *Proper Generalized Decomposition* (PGD) [HEYBERGER, BOUCARD et NÉRON 2012 ; HEYBERGER, BOUCARD et NÉRON 2013].

■ **Formulation du problème multiparamétrique :** Dans la situation présentée ici, on ne s'intéressera qu'à des paramètres d'interface de type *précharge* (imposés à l'aide d'un jeu négatif au niveau de l'interface entre les deux sous-domaines concernés), *jeu* (positif à l'inverse de la précharge) et *coefficients de frottement*.

Ainsi, on regroupe l'ensemble des n_p paramètres considérés sous la forme d'un vecteur $\mathbf{x} = \{x_1, x_2, \ldots, x_{n_p}\}$. L'objectif de la méthode proposée est de pouvoir évaluer à moindre coût la réponse d'un ensemble de n_s jeux de paramètres que l'on notera $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n_s)}\}$.

Au vu du type de paramètres pris en compte et de la formulation de la méthode LATIN, à chaque jeu de paramètres $\mathbf{x}^{(k)}$ ($k \in [\![1, n_s]\!]$), on associe une solution $\mathbf{s}_k$.

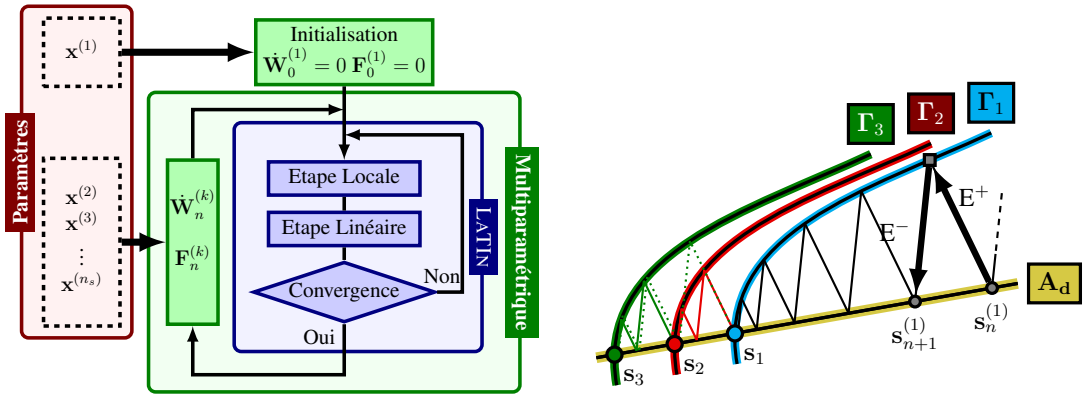
■ **Résolution basée sur la méthode LATIN :** Chaque solution $\mathbf{s}_k = (\mathbf{W}^{(k)}, \mathbf{F}^{(k)})$ associée à un jeu de paramètres est recherchée en employant la méthode LATIN. Ainsi la solution $\mathbf{s}_k$ est obtenue itérativement en cherchant successivement les solutions des sous-espaces $\mathbf{A_d}$ et $\mathbf{\Gamma}_k$. Elle est définie telle que :

$$\mathbf{s}_k \in \mathbf{A_d} \bigcap \mathbf{\Gamma}_k \tag{1.42}$$

Dans la pratique, deux cas sont distingués (figure 1.16a) :

- le calcul de la réponse associée au **premier jeu de paramètres** $\mathbf{x}^{(1)}$ réalisé par l'algorithme LATIN classique où les grandeurs d'interface sont initialisées avec des valeurs nulles au démarrage de l'algorithme ;
- les calculs des réponses associées aux **jeux de paramètres suivants** $\mathbf{x}^{(k)}$ ($k \in [\![2, n_s]\!]$) réalisés par l'algorithme LATIN mais en initialisant les grandeurs d'interface avec des valeurs issues des résolutions précédentes $\mathbf{s}_j$ (avec $j \in [\![1, k-1]\!]$).

■ **Performances** La figure 1.16b présente schématiquement les itérations de l'algorithme LATIN employé avec la stratégie multiparamétrique. Dans le cas présenté, on



(a) Principe de la stratégie multiparamétrique **(b)** Itérations de l'algorithme LATIN pour 3 solutions associées à 3 jeux de paramètres

FIGURE 1.16 – Principe et représentation schématique de la stratégie multiparamétrique basée sur la méthode LATIN

cherche à obtenir les solutions $\mathbf{s}_1$, $\mathbf{s}_2$ et $\mathbf{s}_3$ associées chacune à un jeu de paramètres ($\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$ et $\mathbf{x}^{(3)}$). Une fois la solution $\mathbf{s}_1$ obtenue, le nouveau jeu de paramètres $\mathbf{x}^{(2)}$ permet de construire le sous-espace $\mathbf{\Gamma}_2$. L'algorithme LATIN est initialisé à l'aide des grandeurs d'interface issues de la première résolution. Comme l'illustre la figure 1.16b, la solution $\mathbf{s}_2$ est obtenue en un nombre réduit d'itérations et par conséquent un temps CPU réduit également. L'emploi de la stratégie multiparamétrique permet des réductions significatives des temps de calcul pour ces multiples résolutions (le gain peut aller de 2 à 100 suivant les cas) par rapport à l'emploi d'une méthode LATIN sans stratégie multiparamétrique. Ces performances seront présentées en détail dans le chapitre 2.

■ **Stratégies d'initialisation :** Dans le cas de la figure 1.16b, la solution $\mathbf{s}_3$ peut être obtenue en initialisant l'algorithme LATIN à partir des grandeurs d'interface obtenues lors des précédentes résolutions. Du point de vue du schéma, il semble plus pertinent d'utiliser les grandeurs d'interface de la solution $\mathbf{s}_2$ que de celle de la solution $\mathbf{s}_1$: $\mathbf{s}_2$ permet en effet d'obtenir la solution $\mathbf{s}_3$ en un nombre plus faible d'itérations que la solution $\mathbf{s}_1$ (ligne pointillée sur le schéma). Le choix de la « meilleure » solution d'initialisation [10] est complexe et plusieurs stratégies ont été envisagées [SOULIER et BOUCARD 2009]. Parmi celles-ci, on retrouve :

- l'initialisation par la solution calculée précédemment ;
- l'initialisation par la première solution calculée ;
- l'initialisation par la solution associée au jeu de paramètres le plus proche du jeu considéré. Il est alors nécessaire de définir une norme dans l'espace des paramètres qui permettra ce choix.

10. On définira sous le terme « meilleure » solution d'initialisation, la solution précédemment calculée et utilisée pour initialiser l'algorithme LATIN permettant de minimiser le temps de calcul pour obtenir une solution $\mathbf{s}_k$.

Les premiers résultats proposés dans [SOULIER et BOUCARD 2009] montrent que la troisième stratégie permet d'assurer une réduction importante des temps de calcul lors de l'emploi de la stratégie multiparamétrique (comparé à l'usage de la méthode LATIN sans stratégie multiparamétrique). Des compléments sur le choix d'une stratégie d'initialisation seront proposés dans la partie 2.3.4 page 109 du chapitre 2.

Par ailleurs, la stratégie multiparamétrique et sa mise en œuvre dans le cadre de l'optimisation et pour le calcul des gradients ainsi que des exemples d'assemblages traités avec celle-ci feront l'objet du chapitre 2 page 91.

4 Modèles de substitution

Alors que l'optimisation nécessite pour converger un nombre important d'évaluations de la fonction objectif y et que celles-ci s'avèrent particulièrement coûteuses à obtenir, il n'est pas viable de réaliser une optimisation globale sur cette fonction directement. Une alternative consiste à employer un *modèle de substitution* (*surrogate model*) aussi appelé *métamodèle*. Derrière cette terminologie se cache un ensemble de méthodes capables de fournir une approximation $\tilde{y}$ d'une fonction y obtenue soit de manière analytique soit en utilisant un solveur (ici mécanique). La construction, qui sera détaillée plus loin, nécessite des évaluations de la fonction réalisées en un nombre limité et judicieusement choisi de points de l'espace $\mathcal{D}$ dans lequel varient les paramètres. Une fois construit, le modèle de substitution peut fournir une réponse en un point quelconque de l'espace de conception en un temps de calcul très faible (comparé au temps nécessaire au simulateur mécanique, dans notre cas, pour fournir la réponse exacte au même point).

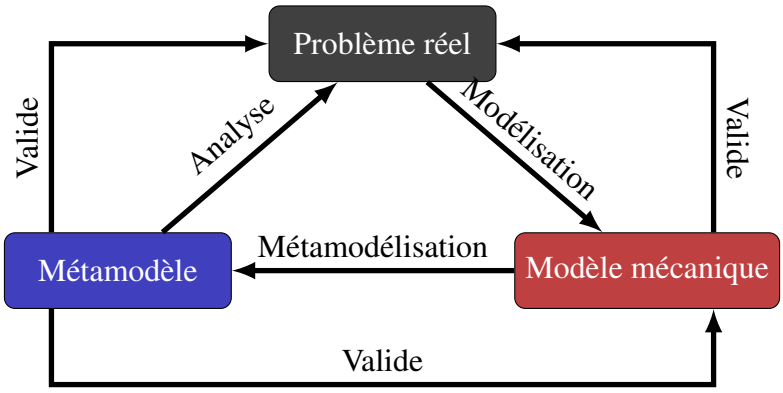


FIGURE 1.17 – Positionnement du métamodèle au sein du contexte de la modélisation du réel issu de [KLEIJNEN et SARGENT 2000]

Comme l'illustre la figure 1.17, le métamodèle s'intègre dans une approche de modélisation du réel. À ce titre, il est important de garder à l'esprit que l'on cherche à construire une approximation permettant une analyse valide du problème réel qui passe par le fait que l'approximation doit assurer une bonne représentativité de la fonction « réelle » obtenue par le simulateur.

Dans cette partie, on s'intéressera plus particulièrement à l'approximation des fonctions continues. Les premières méthodes employées furent les méthodes d'interpolation linéaire, quadratique et polynomiale. Des méthodes d'interpolation polynomiale découle directement la méthodologie des surfaces de réponses introduite par [BOX et WILSON 1951]. Parallèlement à ces techniques d'interpolation, on trouve des méthodes dites de régression qui fournissent une approximation qui ne vérifie pas la valeur des réponses fournies par le modèle exact. La complexité des fonctions que l'on a cherché à approcher augmentant, il a été nécessaire d'augmenter le degré des polynômes employés. Malheureusement, lorsque le degré de ces derniers dépasse 2, de fortes oscillations parasites peuvent être constatées. C'est pour remédier à ce problème et s'adapter à de nouveaux types de fonctions que de nouvelles méthodes ont alors vu le jour.

Dans la suite, le principe de construction d'un modèle de substitution sera rappelé, puis les différentes étapes qui le composent seront évoquées. Quelques méthodes d'échantillonnage de l'espace des paramètres et quelques métamodèles classiques seront en particulier détaillés.

4.1 Notations

Un modèle de substitution nécessite classiquement un échantillon de points et de réponses que l'on nommera **échantillon d'apprentissage** pour être construit. Dans la continuité des notations prises dans les parties précédentes, on désignera par $\mathbf{x}^{(k)}$ un point (ou un site) choisi dans l'espace de conception $\mathcal{D}$ ($\mathcal{D} \subset \mathbb{R}^{n_p}$ où n_p est la dimension du problème). Dans la suite, un point de la base d'apprentissage sera noté $\mathbf{x}^{(k)}$ avec $k \in [\![1, n_s]\!]$ et $\mathbf{x}^{(0)}$ ou $\mathbf{x}$ le point courant de l'espace de conception où l'on souhaite obtenir la valeur approchée de la fonction objectif.

Ainsi, l'échantillon d'apprentissage noté E_a sera défini de la manière suivante :

$$E_a = \{\mathbf{x}^{(1)}, y_1, \mathbf{x}^{(2)}, y_2, \ldots, \mathbf{x}^{(n_s)}, y_{n_s}\} \text{ où } \forall i \in [\![1, n_s]\!], \quad y_i = y\big(\mathbf{x}^{(i)}\big) \tag{1.43}$$

Par ailleurs, on notera ℓ_i un paramètre du métamodèle et l'ensemble de ces paramètres sera regroupé sous la forme d'un vecteur $\mathbf{l}$. Enfin, le modèle de substitution fournit une approximation de la fonction y notée $\tilde{y}$.

4.2 Mise en œuvre d'un modèle de substitution

De manière générale, l'élaboration d'un modèle de substitution nécessite, pour obtenir la réponse en un point, un ensemble E_a de couples de sites et de réponses mais également le choix de fonctions supports et de paramètres. Ces deux derniers éléments sont spécifiques au type de métamodèle et ne sont pas toujours nécessaires. Par ailleurs, en plus de la réponse approchée, certains modèles de substitution fournissent une estimation de qualité globale ou locale (soit sur l'ensemble de l'approximation créée soit en un site donné $\mathbf{x}^{(0)}$). La figure 1.18 présente schématiquement ces éléments.

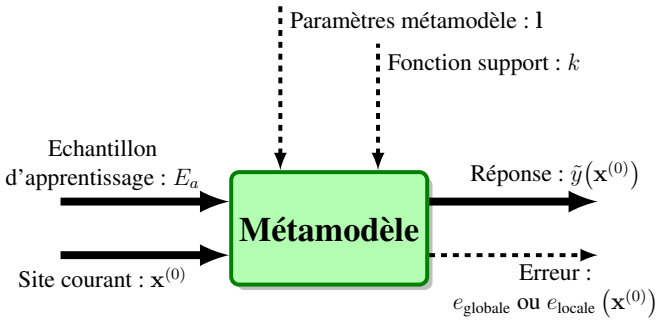


FIGURE 1.18 – Représentation schématique d'un modèle de substitution

Les choix des paramètres du métamodèle et des fonctions support, lorsqu'il y en a, sont des procédures spécifiques à certains métamodèles et nécessitent l'emploi de techniques adaptées.

La procédure de mise en œuvre d'un modèle de substitution est une démarche en plusieurs étapes. Elle fait appel à plusieurs techniques issues de divers champs scientifiques. Parmi celles-ci, on retrouve (cf. figure 1.19 et [FORRESTER, SÓBESTER et KEANE 2008]) :

- une procédure d'échantillonnage permettant d'obtenir un ensemble de points $\mathbf{x}^{(i)}$ de l'espace de conception $\mathcal{D}$. Classiquement, des méthodes de types **plan d'expérience** sont utilisées. Elles seront rappelées dans la partie 4.4 page 66.

- une procédure d'**évaluation des réponses exactes** y_i de la fonction considérée aux points obtenus soit par la stratégie de tirages soit par la procédure d'enrichissement. Cette phase fait appel au modèle mécanique considéré et finalise la construction de la base d'apprentissage E_a.

- la procédure de **construction du métamodèle** a proprement parler. Elle nécessite l'emploi de méthodologies spécifiques au modèle de substitution utilisé. La plupart du temps, lors de la phase de construction, un certain nombre d'opérateurs est construit et ces derniers restent constants lors de l'évaluation du métamodèle (c'est-à-dire pour obtenir $\tilde{y}(\mathbf{x}^{(0)})$ où $\mathbf{x}^{(0)}$ est un point quelconque de $\mathcal{D}$). Cette phase de construction peut s'avérer particulièrement longue pour certains modèles de substitution.

- un test de convergence du modèle de substitution qui se base sur un critère adapté tel que, par exemple, un critère de **validation** mesurant l'écart entre le modèle de substitution et le modèle exact en quelques points. Dans le cas où l'approche considérée intègre une procédure d'enrichissement, ce test peut être employé pour statuer sur la poursuite ou non de l'ajout de nouveaux points pour construire le métamodèle.

- une procédure d'**enrichissement** permettant d'ajouter des nouveaux sites dans la base d'apprentissage. Cet ajout peut être réalisé sur le plan d'expérience existant

(manque de site dans une zone de l'espace de conception) ou en utilisant de façon pertinente les informations fournies par le métamodèle (en employant un indicateur d'enrichissement basé sur la réponse du métamodèle voire un indicateur d'erreur fourni par celui-ci).

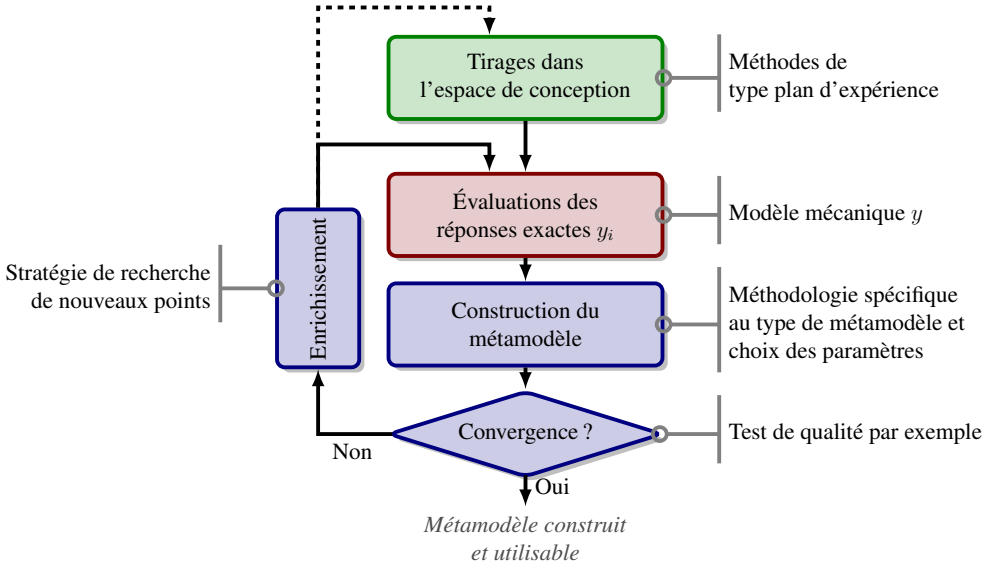


FIGURE 1.19 – Représentation schématique du processus d'élaboration d'un modèle de substitution (adaptée de [FORRESTER, SÓBESTER et KEANE 2008])

Une fois le métamodèle construit, il est alors utilisable et peut fournir à coût extrêmement faible des réponses approchées en n'importe quel point de l'espace de conception.

4.3 Inclusion dans un processus d'optimisation

Au delà de la phase de pure approximation, le modèle de substitution peut avoir une place à part entière dans le processus d'optimisation, c'est-à-dire qu'il doit être pensé et utilisé dans ce but. À la différence de l'approximation pure où l'on va chercher *a priori* à obtenir une approximation de qualité en tout point de l'espace de conception, dans le cadre de l'optimisation, un usage intelligent du métamodèle va permettre de se concentrer sur les zones potentielles où le minimum pourrait se trouver. On parle pour cela de *Surrogate-Based Optimisation* (SBAO) [QUEIPO et al. 2005]. Cette méthode peut être considérée comme une véritable stratégie d'optimisation présentant de fortes similitudes avec des méthodes de région de confiance par exemple [ALEXANDROV, J. E. DENNIS et al. 1998]. La phase d'enrichissement joue pour cela un rôle majeur : elle peut être conçue autour d'un indicateur d'erreur permettant de fournir une information quant aux zones requérant un nouveau point. Il existe un nombre important d'indicateurs [A. G. WATSON et BARNES 1995 ; SASENA 2002] pour l'enrichissement parmi lesquels on trouve l'amélioration espérée (*Expected Improvement*) et ses dérivées [SCHONLAU

1997 ; SÓBESTER, LEARY et KEANE 2005] formant la base de l'algorithme d'optimisation globale connu sous le nom EGO (*Efficient Global Optimisation*) [D. R. JONES, SCHONLAU et WELCH 1998]. Les différentes méthodes d'enrichissement seront détaillées dans le chapitre 4 page 201.

4.4 Méthodes d'échantillonnage de l'espace des paramètres

La procédure de construction des métamodèles se base sur un ensemble de sites $\mathbf{x}^{(i)}$ choisis dans l'espace de conception et des réponses y_i associées. L'obtention des échantillons est une problématique de recherche à part entière. Initialement introduites pour le traitement de données expérimentales, les méthodes dites de *plans d'expérience* (ou *Design of Experiments*) [BOX, J. S. HUNTER et W. G. HUNTER 1978 ; TAGUCHI 1986 ; MYERS et MONTGOMERY 1995] permettent, dans un cadre expérimental, de minimiser le nombre de manipulations tout en étant capable d'évaluer ou tout du moins de contrôler l'erreur dite de répétabilité [11], de quantifier l'influence d'un paramètre sur la réponse du modèle ainsi que celle de l'interdépendance entre les paramètres. Dans le contexte de cette thèse, on ne s'intéressera qu'à des **réponses déterministes** du modèle. Ainsi, il ne sera pas tenu compte de la variabilité de la réponse fournie par le modèle exact.

L'objectif de ces plans d'expériences numériques sera alors de proposer un échantillonnage de l'espace de conception qui permette d'obtenir une bonne représentabilité de la fonction que l'on souhaite approximer. Les plans d'expérience classiques ne sont généralement pas conçus pour tenir compte de la complexité des modèles actuels. Ils sont en effet dédiés à la modélisation de la fonction par des polynômes de premier et second degré et concentrent les points sur le bord de l'espace d'étude. C'est principalement pour remédier à ces défauts et s'adapter à la problématique de la simulation numérique de données issues de codes de calcul que les plans d'expériences « numériques » ont été envisagés [SACKS, WELCH et al. 1989] (cf. par exemple [FRANCO 2008] pour une synthèse).

Dans le but de tirer le maximum d'informations du modèle considéré, les échantillons doivent (cf. [FRANCO 2008]) :

- approcher l'orthogonalité des différents paramètres de conception ;
- remplir l'espace de conception, c'est-à-dire que les échantillons doivent former un sous-ensemble représentatif de l'espace où varient les paramètres ;
- être obtenus de manière efficace même lorsque le nombre de paramètres est important ;
- assurer l'obtention d'un maximum d'effets d'interactions ;
- être obtenus facilement.

11. On parle d'erreur de répétabilité pour désigner l'écart entre les réponses du modèle expérimental pour un même jeu de paramètres

La construction de métamodèles pour l'optimisation s'écartant du cadre de l'étude des interactions entre les paramètres, on cherchera à employer une méthode d'échantillonnage qui assure une bonne couverture de l'espace de conception mais également une bonne uniformité afin d'être capable de construire un modèle de substitution représentatif de la fonction réelle.

Il est possible de classer les méthodes d'échantillonnage existantes en 3 catégories :

- les méthodes de remplissage de l'espace ;
- celles basées sur des critères statistiques ;
- celles basées sur des critères déterministes.

Remarque 1.10 : La présentation des plans d'expérience qui suit n'est pas exhaustive mais tend à donner un aperçu des méthodes existantes. Pour des compléments, on pourra se référer à [SACKS, WELCH et al. 1989 ; BATES et al. 1996 ; KOEHLER et OWEN 1996 ; JOURDAN 2000 ; T. W. SIMPSON, LIN et W. CHEN 2001 ; FRANCO 2008].

4.4.1 Rappels sur les plans d'expérience usuels

Dans cette partie, les plans d'expérience « standards » introduits principalement pour des applications de type surfaces de réponse seront brièvement rappelés. Parmi ceux-ci, on retrouve :

- les **plans factoriels complets** (cf. figure 1.20) qui consistent à remplir l'espace de conception à l'aide d'une grille régulière. Ces plans présentent l'avantage d'offrir un très bon remplissage de l'espace. Ce remplissage n'est cependant viable que pour de faibles dimensions (2 ou 3). En effet, le nombre d'échantillons n_s augmente rapidement avec le nombre de niveaux k choisis par paramètres et la dimension n_p du problème ($n_s = k^{n_p}$). Il peut par ailleurs conduire à de nombreuses informations redondantes lorsque la fonction y est indépendante de certains paramètres.

- les **plans factoriels fractionnaires** (cf. figure 1.21) consistant à prendre des sous-ensembles de plans factoriels complets. Ils permettent une réduction du nombre de simulations mais des problèmes d'alignement persistent.

- les **plans composites** (cf. figure 1.22) qui cherchent à améliorer certaines propriétés des plans factoriels à 2 niveaux. Leur construction s'appuie sur un plan factoriel complet ou fractionnaire et ajoute de nouveaux points. Ces plans d'expérience ont été proposés pour approcher des surfaces de réponse de degré 2. Cependant, ils ne présentent ni un bon remplissage de l'espace, ni surtout une bonne répartition des points. Par ailleurs, le nombre de simulations augmente de façon importante avec le nombre de paramètres.

- les **plans de Box-Behnken** (cf. figure 1.23) où chaque paramètre ne peut prendre que 3 valeurs (3 niveaux). Les tirages obtenus correspondent aux milieux des arêtes de l'hypercube formé par l'espace des paramètres $\mathcal{D}$. À ces points vient s'ajouter le point central de l'hypercube.

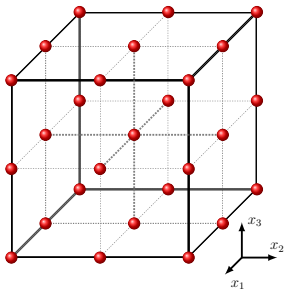


FIGURE 1.20 – Plan factoriel complet 3^3 (27 points)

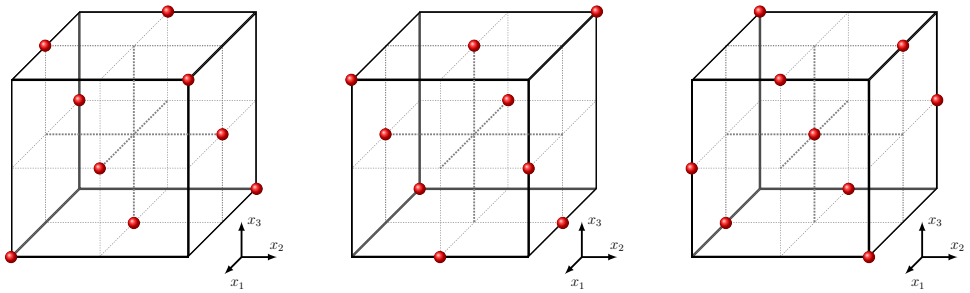


FIGURE 1.21 – Exemples de plans fractionnaires issus du plan complet 3^3

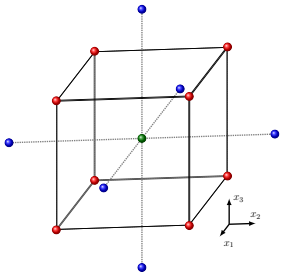


FIGURE 1.22 – Plan composite centré pour 3 paramètres et 2 niveaux

Il existe de nombreux autres plans d'expériences utilisables principalement pour des approches de type surface de réponse. Le lecteur souhaitant des compléments d'informations concernant ces plans d'expérience pourra se référer aux ouvrages [MYERS et MONTGOMERY 1995 ; G. A. LEWIS, MATHIEU et PHAN-TAN-LUU 1999].

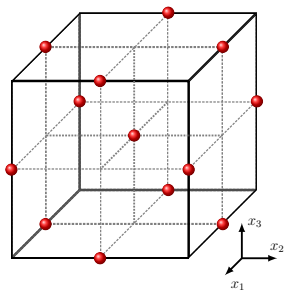


FIGURE 1.23 – Plan de Box-Behnken pour 3 paramètres

Les plans d'expérience présentés brièvement dans cette partie assurent pour la plupart de bonnes performances en termes de remplissage de l'espace. Malheureusement, ils sont particulièrement coûteux en nombre de simulations qu'ils engendrent. C'est pourquoi il est préférable de se tourner vers des stratégies d'échantillonnage moins usuelles.

4.4.2 Méthodes de remplissage de l'espace (*Space Filling Design*)

Dans cette partie, quelques méthodes dites de « remplissage de l'espace » (*Space Filling Design*) seront rappelées. L'idée est ici de chercher à trouver un compromis entre le nombre de sites fournis par la méthode, le remplissage de l'espace et l'uniformité de ces sites.

■ **Méthode Hypercube Latin (LHS) :** Cette méthode [MCKAY, CONOVER et BECKMAN 1979] initialement introduite pour calculer numériquement des intégrales multiples présente la particularité de fournir des échantillons non redondants et assez bien répartis. L'objectif était d'améliorer les propriétés de convergence de méthodes de Monte-Carlo nécessitant un nombre important de tirages pour parvenir à converger vers une valeur correcte de l'intégrale considérée. D'une grande simplicité d'usage et de construction, elle est actuellement énormément utilisée (par exemple pour la construction des modèles de substitution de type krigeage [VIANA 2013]).

La méthode LHS consiste à découper en n_s segments de même longueur chacun des n_p axes $[b_i^l, b_i^u]$ (tel que $\forall i \in [\![1, n_p]\!]$, $b_i^l \leq x_i \leq b_i^u$). Ce découpage permet l'obtention d'un maillage comportant $n_s^{n_p}$ cellules parmi lesquelles n_s sont choisies de manière à ce qu'en projection sur chacun des axes $[b_i^l, b_i^u]$ chaque cellule soit sélectionnée une seule fois. Finalement, un tirage aléatoire permet l'obtention d'un point dans chacune de ces n_s cellules (cf. figure 1.24).

Les points $\mathbf{x}^{(j)}$ ($j \in [\![1, n_s]\!]$) d'un hypercube $[0,1]^{n_p}$ sont obtenus de la manière suivante

$$\forall i \in [\![1, n_p]\!],\ \forall j \in [\![1, n_s]\!],\ x_i{}^{(j)} = \frac{\mathbf{\Pi}_i(j) - U_i^{(j)}}{n_s} \tag{1.44}$$

où $\mathbf{\Pi}_i(j)$ est une permutation de $[\![1, n_s]\!]$ et U_j^i est une valeur aléatoire de distribution uniforme sur $[0,1]$. Ainsi, les ensembles $(\mathbf{\Pi}_1(j), \ldots, \mathbf{\Pi}_{n_p}(j))$ et $(U_1^{(j)}, \ldots, U_{n_p}^{(j)})$ représentent respectivement la cellule dans laquelle se trouve le point et l'endroit où se situe le point dans la cellule.

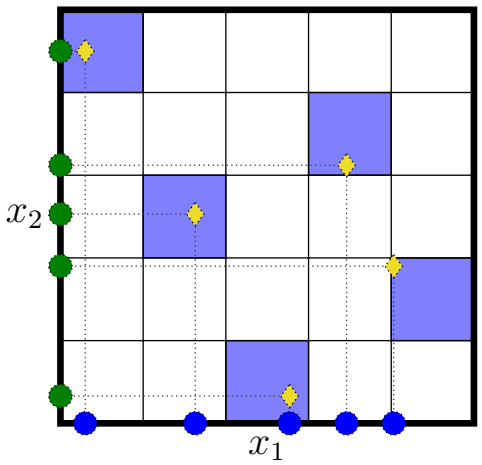
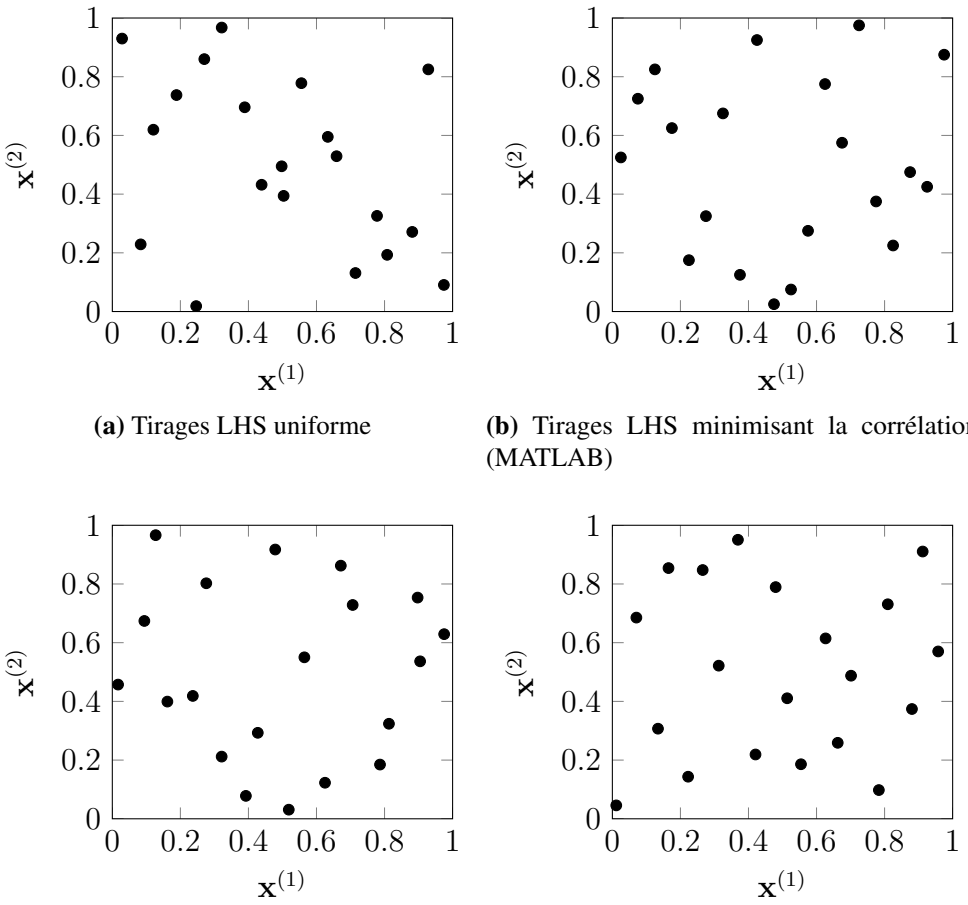


FIGURE 1.24 – Exemple de 5 sites obtenus par LHS

Dans le cas proposé en figure 1.24, les sites obtenus ont la propriété d'être uniformément distribués selon chaque axe $[b_i^l, b_i^u]$. Cependant, les permutations retenues n'assurent pas nécessairement l'uniformité des sites sur l'espace de conception. Ainsi lors de l'usage de la méthode LHS pour un faible nombre de sites, il est courant d'observer des zones sans points.

Afin de remédier à ces problèmes, on peut envisager soit de générer plusieurs tirages LHS et de choisir le meilleur en utilisant un critère soit de construire un tirage LHS et de l'améliorer en employant un critère et une stratégie adaptés. Parmi ces stratégies de construction, on trouve celles destinées à construire un hypercube latin orthogonal en améliorant un critère de corrélation [TANG 1993 ; OWEN 1994 ; TANG 1998], celles visant à minimiser un critère statistique tel que l'entropie [M. D. MORRIS et MITCHELL 1995], le critère IMSE (*Integrated Mean Square Error*) [PARK 1994] ou celles permettant de sélectionner un tirage LHS ayant de bonnes propriétés d'uniformité en employant des critères tels que les critères de distance optimale entre les sites (IHS - *Improved Hypercube Sampling*) [BEACHKOFSKI et GRANDHI 2002], des critères *maximin* [12] ou *minimax* [13] [JOHNSON, MOORE et YLVISAKER 1990].

12. le critère *maximin* maximise la distance entre deux points de l'échantillon
13. le critère *minimax* minimise la distance maximale entre un point du domaine et un point de l'échantillon



(a) Tirages LHS uniforme

(b) Tirages LHS minimisant la corrélation (MATLAB)

(c) Tirages LHS avec critère *maximin* (R)

(d) Tirages IHS (R)

FIGURE 1.25 – Quelques exemples de tirages de type *Latin Hypercube Sampling* en dimension 2 pour $n_s = 20$

Remarque 1.11 : Le logiciel (et langage) de calculs statistiques R [a] intègre plusieurs routines permettant d'obtenir les différents tirages de type LHS ainsi que d'ajouter des sites à un tirage existant. On y retrouve le tirage LHS de base, des tirages LHS respectant des critères de S-optimalité, de *maximin* et de distance euclidienne optimale entre deux points. Ces routines sont regroupées au sein du package `lhs` [b]. Le package `DiceDesign` [c] comporte également quelques routines permettant l'obtention de tirages LHS « optimaux ».

a. `http://cran.r-project.org/`
b. `http://cran.r-project.org/web/packages/lhs/index.html`
c. `http://cran.r-project.org/web/packages/DiceDesign/index.html`

Remarque 1.12 : De la même manière que pour le logiciel R, il est possible de trouver des routines MATLAB assurant les constructions de ce type de tirages LHS. La toolbox *statistics* comprend la fonction `lhsdesign` [a] permettant d'obtenir un tirage LHS avec ou sans critère d'amélioration (critère *maximin* ou basé sur la *corrélation*). On trouve également des toolbox librement accessibles telle que la toolbox *Latin Hypercube Sampling* [b] ou IHS proposant notamment d'optimiser l'espace entre deux points [c].

a. `http://www.mathworks.fr/fr/help/stats/lhsdesign.html`
b. `http://www.mathworks.com/matlabcentral/fileexchange/4352-latin-hypercube-sampling`
c. `http://people.sc.fsu.edu/~jburkardt/m_src/ihs/ihs.html`

■ **Tableaux orthogonaux** Les tableaux orthogonaux [TAGUCHI 1986 ; OWEN 1992] sont des plans d'expériences proches des plans de type hypercube latin. Leur construction est similaire à celle des LHS : elle repose sur un partitionnement de chaque axe de l'espace de conception en q segments égaux afin d'obtenir un maillage de l'espace constitué de q^{n_p} cellules. On choisit n_s cellules telles qu'elles forment un tableau orthogonal c'est-à-dire respectant la définition.

Définition 1.4 (Tableau orthogonal) **:** Un tableau orthogonal de force t à q symboles est une matrice de n_s lignes et n_p colonnes avec ses éléments prenant q valeurs possibles, telles que toutes les sous-matrices de taille $n_s \times t$ aient q^t lignes apparaissant le même nombre λ fois ($n_s = \lambda \times q^t$). Un tel tableau orthogonal sera noté $\mathrm{OA}_\lambda(n_s, n_p, q, t)$.

À l'instar des LHS, les points sont choisis aléatoirement dans chacune des n_s cellules.

Un tableau orthogonal comportant n_s sites $\mathbf{x}^{(j)}$ ($j \in [\![1, n_s]\!]$) d'un hypercube $[0,1]^{n_p}$ sont obtenus de la manière suivante

$$\forall i \in [\![1, n_p]\!],\ \forall j \in [\![1, n_s]\!],\ x_i{}^{(j)} = \frac{\mathbf{\Pi}_i\left(A_j^{(i)}\right) - U_i^{(j)}}{q} \tag{1.45}$$

où $\mathbf{\Pi}_i(j)$ est une permutation de $[\![1, q-1]\!]$, $A_j^{(i)}$ sont les éléments du tableau orthogonal et U_j^i est une valeur aléatoire de distribution uniforme sur $[0,1]$. Ainsi, les ensembles

$(\mathbf{\Pi}_1\left(A_1^{(i)}\right),\ldots,\mathbf{\Pi}_{n_p}\left(A_{n_p}^{(i)}\right))$ et $(U_1^{(j)},\ldots,U_{n_p}^{(j)})$ représentent respectivement la cellule dans laquelle se trouve le point et l'endroit où se situe le point dans la cellule.

Des compléments concernant les plans orthogonaux et leur élaboration peuvent être trouvés dans [JOURDAN 2000 ; FRANCO 2008]. L'un des principaux défauts de ces types d'échantillonnage réside dans le fait qu'ils engendrent un nombre de simulations et par conséquent un coût importants.

■ **Méthodes basées sur les suites à discrépance faible :** Les précédents plans d'expérience présentent de bonnes répartitions en projection sur les axes de l'espace où varient les paramètres. Les sites obtenus ne sont cependant pas forcément bien répartis dans l'espace. Pour remédier à ça, il est possible d'utiliser des suites dites à faible *discrépance* permettant d'obtenir des points répartis de manière plus uniforme dans l'espace de conception.

La discrépance est une mesure de l'écart entre les sites obtenus et une distribution de points uniforme. Elle ne sera pas développée ici mais des éléments concernant l'évaluation de la discrépance peuvent être trouvés dans [NIEDERREITER 1987 ; THIÉMARD 2000].

Parmi les suites à faible discrépance classiquement employées pour réaliser un échantillonnage, on trouve les suites de Hammersley [HAMMERSLEY 1960 ; KALAGNANAM et DIWEKAR 1997], de Halton [HALTON 1960 ; RAFAJŁOWICZ et SCHWABE 2006] et de Sobol [SOBOL 1967].

Remarque 1.13 : Le package `DiceDesign`[a] du logiciel R comporte quelques routines permettant l'évaluation des discrépances « classiques ».

a. `http://cran.r-project.org/web/packages/DiceDesign/index.html`

■ **Uniform Design :** En lieu et place des suites à discrépance faible pour générer l'ensemble des sites, il est possible d'utiliser un algorithme d'optimisation. Le positionnement des points pourra alors se faire en cherchant à minimiser un critère de discrépance [FANG 1980 ; Y. WANG et FANG 1981 ; FANG et al. 2000]. Le coût de construction de ce type de plans peut devenir extrêmement important car il nécessite la résolution d'un problème d'optimisation de taille $n_s \times n_p$. C'est pour cette raison que l'on préfère généralement construire plusieurs ensembles de n_s tirages pour choisir ensuite celui minimisant un critère que l'on choisit.

■ **Critères déterministes :** La construction de plans d'expériences « optimaux » au sens d'un critère choisi peut être réalisée en considérant un certain nombre de métriques. Celles-ci évaluent les distances intersites et/ou les distances entre les sites et une grille régulière. On parle alors de critères d'*uniformité*. Parmi les critères classiques, on retrouve pour un ensemble de points échantillonnés dans le cube unité $[0,1]^{n_p}$ [GUNZBURGER et BURKARDT 2004] :

- une **mesure de recouvrement** (*COV measure*) désignée par λ et définie telle que

$$\forall i \in [\![1, n_s]\!],\ \forall \mathbf{x}^{(i)} \in \mathcal{D},\ \lambda = \frac{1}{\overline{\gamma}} \sqrt{\frac{1}{n_s} \sum_{i=1}^{n_s} (\gamma_i - \overline{\gamma})^2} \tag{1.46}$$

  où $\gamma_i = \min_{k \geq i} \|\mathbf{x}^{(i)}, \mathbf{x}^{(k)}\|_2$ désigne la distance minimale entre le point $\mathbf{x}^{(i)}$ et les autres points du tirage et $\overline{\gamma} = \frac{1}{n_s} \sum_{i=1}^{n_s} \gamma_i$ la moyenne des γ_i. Un tirage se rapprochant d'une grille régulière verra sa mesure de recouvrement se rapprocher de 0.

- le **rapport de distances** (*mesh ratio*) désignées par R et défini tel que

$$\forall i \in [\![1, n_s]\!],\ \forall \mathbf{x}^{(i)} \in \mathcal{D},\ R = \frac{\max\limits_{i \in [\![1,n_s]\!]} \gamma_i}{\min\limits_{i \in [\![1,n_s]\!]} \gamma_i} \tag{1.47}$$

  où γ_i est défini de la même manière que pour la mesure de recouvrement. Un tirage se rapprochant d'une grille régulière verra son rapport de distances se rapprocher de 1.

- les critères **maximin** et **minimax** [JOHNSON, MOORE et YLVISAKER 1990] permettant respectivement l'obtention de plans qui

  - maximisent la distance minimum entre deux points de l'échantillon :

$$\max_{i \in [\![1,n_s]\!]} \min_{j \in [\![1,n_s]\!]} \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_2 \tag{1.48}$$

  - minimisent la distance maximale entre un ensemble $\mathcal{E}$ de points de l'espace $\mathcal{D}$ et les points obtenus par la stratégie d'échantilonnage :

$$\min_{\mathbf{x} \in \mathcal{E}} \max_{i \in [\![1,n_s]\!]} \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_2 \tag{1.49}$$

    Généralement pour calculer le critère *minimax*, les points d'une grille régulière sont utilisés mais sa mise en œuvre peut s'avérer complexe. Le critère *maximin*, plus simple à mettre en œuvre, a le défaut de concentrer les points sur le bord de l'espace.

La mise en œuvre de stratégies dédiées à l'amélioration de ces critères peut s'avérer complexe et nécessite d'utiliser des règles de permutation et de déplacement des sites (cf. [FRANCO 2008] pour quelques algorithmes et références concernant ces méthodes).

■ **Critères statistiques :** En complément des critères déterministes, il est également possible d'employer des critères statistiques pour construire les plans. Ils permettent par exemple de tenir compte d'informations connues sur les paramètres que l'on souhaite échantillonner et/ou de s'adapter au type de modèle de substitution que l'on souhaite construire. Parmi les critères statistiques, on retrouve

- le critère d'entropie [LINDLEY 1956] permettant l'obtention de plans dits « à entropie maximale » [SHEWRY et WYNN 1987]. La prise en compte de ce critère basé sur la notion d'entropie de Shannon [SHANNON 1948] a pour but de maximiser la quantité d'informations apportées par les simulations relativement à un paramètre du modèle. Ce critère ne peut donc se concevoir sans un modèle de prédiction *a posteriori* (comme du krigeage par exemple).
- l'erreur moyenne quadratique intégrée (IMSE *Integrated Mean Square Error*), aussi connue sous le nom de *critère de "variance+biais"* [BOX et DRAPER 1959], permettant de sélectionner le plan qui minimise l'écart entre la réponse du simulateur et sa prédiction [WELCH 1983 ; SACKS, SCHILLER et WELCH 1989 ; SACKS, WELCH et al. 1989].

Ces approches sont particulièrement lourdes à mettre en œuvre. En particulier, elles requièrent un coût numérique important, d'autant plus qu'elles sont dépendantes du modèle de prédiction employé et nécessitent une reconstruction du plan lorsqu'un paramètre du modèle change.

Remarque 1.14 : Les plans basés sur des critères statistiques sont également connus sous un ensemble de dénominations [GAUCHI 2005] : on retrouve les critères A-optimaux minimisant la trace de la matrice de covariance, D-optimaux minimisant le déterminant de la matrice de covariance, E-optimaux minimisant la valeur propre maximale de la matrice de covariance, G-optimaux minimisant le maximum de la variance prédite (par un modèle de substitution par exemple), J-optimaux minimisant le critère IMSE, S-optimaux minimisant l'entropie...

4.4.3 Bilan des stratégies de tirages

Comme nous l'avons brièvement vu ici, le nombre et la variété de stratégies d'échantillonnage sont extrêmement importants. Les stratégies d'échantillonnage forment à elles seules un véritable champ de prospection scientifique qui ne peut être pleinement envisagé dans les travaux présentés ici. On se propose alors d'employer dans la suite de ces travaux des tirages obtenus par LHS et LHS optimaux pour des raisons de simplicité de mise en œuvre et de disponibilité des outils.

4.5 Modèles de substitution

Une fois le tirage de sites de l'espace de conception et les réponses associées obtenues, il est alors temps de construire un métamodèle ou modèle de substitution, c'est-à-dire un

modèle capable d'approcher le comportement d'une fonction parfois coûteuse à obtenir. Le nombre de métamodèles existants dans la littérature est extrêmement important. On peut cependant les classer en deux catégories :

- les **métamodèles paramétriques** qui, comme leur nom l'indique, nécessitent de choisir des paramètres internes de manière manuelle par un choix plus ou moins empirique de l'utilisateur ou de manière « automatique » en mettant en œuvre une méthodologie destinée à les fixer. Pour certains métamodèles paramétriques, le choix précis de ces paramètres est indispensable et ne peut être ignoré. Ces techniques, souvent spécifiques au type de métamodèles employés, nécessitent des outils adaptés tels que, par exemple, un algorithme d'optimisation. Cette phase de choix du ou des paramètres peut accaparer la majorité du coût nécessaire à la construction du modèle de substitution. Parmi les modèles paramétriques, on trouve les modèles de type krigeage, RBF (*Radial Basis Functions*), les réseaux de neurones ou perceptrons multi-couches, les Machines à Vecteur de Support (*Support Vector Machine*).
- **métamodèles non-paramétriques** qui, à l'inverse des précédents, ne requièrent aucune détermination de paramètres internes. Historiquement, c'est ce type de modèles qui furent les premiers utilisés avec par exemple la régression linéaire, quadratique et polynomiale.

Dans cette partie, on se propose de présenter un bref état de l'art des métamodèles existants dans la littérature.

Remarque 1.15 : Il a été choisi de **ne pas introduire ici** les métamodèles de type **Krigeage** et basés sur les **fonctions de bases radiales**. En effet, ceux-ci et leur emploi feront l'objet d'une explication détaillée dans le cadre de l'interpolation basée sur les réponses et les **gradients** de la fonction étudiée dans la partie 3 page 160 du chapitre 3.

4.5.1 Historique des modèles de substitution

Les approches d'approximation de fonctions voient le jour dans les années 1950. La méthode des surfaces de réponse (*Response Surface Methodology*) fut élaborée comme outil d'aide aux manipulations expérimentales et de dépouillement de résultats. S'appuyant sur des régressions polynomiales de second degré au maximum, la méthode des surfaces de réponses ne permet pas d'approcher des fonctions non-linéaires plus complexes. C'est pourquoi il est nécessaire de se tourner vers d'autres méthodes.

C'est dès 1943 que l'idée des réseaux de neurones apparaît. Proposée par [McCulloch et Pitts 1943], l'idée repose sur une modélisation mathématique du comportement des neurones. A ces premiers travaux viendront s'ajouter, au fur et à mesure des années, une règle d'apprentissage [Hebb 1949], le premier réseau perceptron comportant une couche de perceptron (les entrées) et une couche de décision (la sortie) [Rosenblatt 1958], ce qui forme le premier réseau capable d'apprendre par expérience.

Parallèlement aux réseaux de neurones, le krigeage a été introduit dans un contexte d'analyses statistiques de données minières [Krige 1951a ; Krige 1951b]. Le nom de

krigeage (*kriging*) fut par la suite proposé par [MATHERON 1963b] qui en réalisa les fondements mathématiques dans un cadre géo-statistique. Des développements supplémentaires furent réalisés dans les années 1990 par [CRESSIE 1993]. C'est en 1989 que le krigeage fut utilisé dans le domaine de l'optimisation de forme [SACKS, SCHILLER et WELCH 1989].

Dans les années 1980, [FRIEDMAN 1991] propose la régression multivariée par splines adaptatives (*Multivariate Adaptative Regression Splines* - MARS). On retrouve également les réseaux de fonctions de base radiale (*Radial Basis Function* - RBF) qui sont une forme particulière des réseaux de neurones [BROOMHEAD et LOWE 1988].

Dans les années 1990 ont également vu le jour les Machines à Vecteur Support (*Support Vector Machine* - SVM), qui semblent prometteuses en raison de leur cadre théorique puissant [VAPNIK 1995].

4.5.2 Régression polynomiale

Les modèles à régression polynomiale s'inscrivent dans une démarche de modélisation par surface de réponse (*Response Surface Methodology* (RSM) [BOX et WILSON 1951]). Cette technique était originellement destinée à analyser des résultats d'expériences physiques, à créer des modèles empiriques à partir des valeurs observées et à interpréter statistiquement les résultats [BOX et DRAPER 1987 ; MYERS et MONTGOMERY 1995]. Largement employée en raison de son faible coût de mise en œuvre et du caractère explicite de la fonction approchée, elle est relativement simple à mettre en œuvre.

On se limitera dans cette partie à présenter le principe de la régression polynomiale dans un cadre déterministe simplement comme outil d'approximation.

Dans le cadre de la régression polynomiale, on cherche l'approximation $\tilde{y}$ de la fonction y sous la forme :

$$\tilde{y}(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n_p} \beta_i x_i + \sum_{i=1}^{n_p} \sum_{i \leq j}^{n_p} \beta_{ij} x_i x_j \tag{1.50}$$

où β_i et β_{ij} ($\forall i \in [\![1, n_p]\!]$ et $i < j \leq n_p$) sont les $n_t = (n_p + 1)(n_p + 2)/2$ coefficients inconnus à déterminer.

L'équation 1.50 donne la formulation de l'approximation de la régression polynomiale de degré deux.

Remarque 1.16 : Il est également possible de formuler l'approximation sous forme vectorielle $\tilde{y}(\mathbf{x}) = \bar{\mathbf{f}}\boldsymbol{\beta}$ où $\boldsymbol{\beta} = \begin{bmatrix}\beta_0 & \beta_1 & \dots & \beta_{n_p} & \beta_{11} & \beta_{12} & \dots & \beta_{n_p n_p}\end{bmatrix}^T$ et $\bar{\mathbf{f}} = \begin{bmatrix}1 & x_1 & \dots & x_{n_p} & x_1^2 & x_1 x_2 & \dots & x_{n_p}^2\end{bmatrix}$.

La détermination des n_t coefficients nécessite au moins autant de couples $\left\{\mathbf{x}^{(i)}, y_i\right\}$ distincts. Dans la pratique, on dispose généralement d'un nombre de mesures supérieur au nombre de coefficients. La détermination des coefficients se fait alors par la méthode des moindres carrés.

On considère un échantillon composé de n_s couples $\left\{\mathbf{x}^{(i)}, y_i\right\}$. On peut alors formuler l'erreur d'approximation pour chaque couple sous la forme $\varepsilon_i = \tilde{y}\left(\mathbf{x}^{(i)}\right) - y\left(\mathbf{x}^{(i)}\right)$. Ainsi sous forme vectorielle et en posant

$$\mathbf{F} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \dots & x_{n_p}^{(1)} & \left(x_1^{(1)}\right)^2 & x_1^{(1)}x_2^{(1)} & \dots & \left(x_{n_p}^{(1)}\right)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(n_s)} & x_2^{(n_s)} & \dots & x_{n_p}^{(n_s)} & \left(x_1^{(n_s)}\right)^2 & x_1^{(n_s)}x_2^{(n_s)} & \dots & \left(x_{n_p}^{(n_s)}\right)^2 \end{bmatrix}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 & \beta_1 & \dots & \beta_{n_p} & \beta_{11} & \beta_{12} & \dots & \beta_{n_p n_p} \end{bmatrix}^T$$

$$\mathbf{y}_s = \begin{bmatrix} y\left(\mathbf{x}^{(1)}\right) & \dots & y\left(\mathbf{x}^{(n_s)}\right) \end{bmatrix}^T$$

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 & \dots & \varepsilon_{n_s} \end{bmatrix}^T$$

l'erreur d'approximation s'écrit

$$\boldsymbol{\varepsilon} = \mathbf{F}\boldsymbol{\beta} - \mathbf{y}_s \tag{1.51}$$

La méthode des moindres carrés consiste ensuite à chercher $\boldsymbol{\beta}$ minimisant la norme de l'erreur quadratique (*Mean Square Error* (MSE)) suivante :

$$\text{MSE}\left(\boldsymbol{\beta}\right) = \sum_{i=1}^{n_s} \varepsilon_i^2 = \left\|\mathbf{F}\boldsymbol{\beta} - \mathbf{y}_s\right\|_2^2 = \boldsymbol{\beta}^T\mathbf{F}^T\mathbf{F}\boldsymbol{\beta} - 2\boldsymbol{\beta}^T\mathbf{F}^T\mathbf{y}_s + \mathbf{y}_s^T\mathbf{y}_s \tag{1.52}$$

Le gradient de la fonction MSE($\boldsymbol{\beta}$) est égal à $\nabla\text{MSE} = 2\mathbf{F}^T(\mathbf{F}\boldsymbol{\beta} - \mathbf{y}_s)$. Ce qui conduit, en utilisant la convexité de la fonction, au vecteur $\hat{\boldsymbol{\beta}}$ minimisant la fonction :

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{F}^T\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{y}_s \tag{1.53}$$

L'approximation polynomiale est à la fois facile d'utilisation et d'implémentation. Elle est cependant d'une précision faible lorsque la fonction à approcher est multimodale [14] suivant certaines composantes. Une montée en degré permet de rendre l'approximation plus précise mais peut entraîner, dans certains cas, l'apparition d'oscillations parasites artificielles. Un des inconvénients de cette méthode réside dans le coût numérique qu'elle engendre lorsque les dimensions deviennent importantes (nombre de paramètres et de points échantillonnés pris en compte). Des compléments concernant l'usage de la régression polynomiale pour l'optimisation peuvent être trouvés dans [VENTER, HAFTKA et STARNES 1998].

En optimisation, les capacités de lissage de la technique de régression polynomiale assurent une convergence rapide pour des fonctions particulièrement bruitées [GIUNTA, DUDLEY et al. 1994].

Enfin, lorsque les dérivées de la fonction à approcher sont connues, il est possible d'utiliser une méthode d'approximation multivariable de Hermite [L. WANG, GRANDHI et CANFIELD 1996].

14. On parle de fonction multimodale pour désigner une fonction possédant plusieurs extrema.

4.5.3 Moindres carrés mobiles

L'approche des moindres carrés mobiles (*Moving Least Square* - MLS) [LANCASTER et SALKAUSKAS 1981 ; LANCASTER et ŠALKAUSKAS 1986] est devenue populaire suite à son utilisation pour construire des fonctions de forme dans le cadre de méthodes sans maillage pour résoudre des problèmes d'équations aux dérivées partielles. L'idée repose sur la construction d'une approximation de la fonction y à partir de n_s couples $\{\mathbf{x}^{(i)}, y\big(\mathbf{x}^{(i)}\big)\}$ sous la forme

$$\forall \mathbf{x}^{(0)} \in \mathcal{D},\ \tilde{y}\big(\mathbf{x}^{(0)}\big) = \mathbf{p}^T\left(\mathbf{x}^{(0)}\right)\boldsymbol{\beta}\left(\mathbf{x}^{(0)}\right) \tag{1.54}$$

Classiquement, les fonctions de base $\mathbf{p}$ sont choisies telles que

$$\mathbf{p}\left(\mathbf{x}\right) = \begin{bmatrix}1 & x_1 & x_2 & \dots & x_{n_p} & x_1^2 & x_1x_2 & \dots & x_{n_p}^2\end{bmatrix} \tag{1.55}$$

La différence de cette approche avec la régression polynomiale classique réside dans le fait que les coefficients β_i et β_{ij} sont dépendants du point $\mathbf{x}^{(0)}$ où l'on cherche l'approximation. Leur détermination est réalisée en employant une méthode de type *moindres carrés pondérés*. On cherche alors les coefficients β_i et β_{ij} qui minimisent l'erreur suivante :

$$E_{WLS}\left(\mathbf{x}^{(0)}\right) = \sum_{i=1}^{n_s} w\left(\|\mathbf{x}^{(i)} - \mathbf{x}^{(0)}\|\right)\left(\mathbf{p}^T\left(\mathbf{x}^{(i)} - \mathbf{x}^{(0)}\right)\boldsymbol{\beta}\left(\mathbf{x}^{(0)}\right) - y\big(\mathbf{x}^{(i)}\big)\right) \tag{1.56}$$

La minimisation de l'erreur E_{WLS} conduit à

$$\forall \mathbf{x}^{(0)} \in \mathcal{D},\ \mathbf{a}\left(\mathbf{x}^{(0)}\right) = \mathbf{A}\left(\mathbf{x}^{(0)}\right)^{-1}\mathbf{B}\left(\mathbf{x}^{(0)}\right)\mathbf{y}_s \tag{1.57}$$

où

$$\mathbf{A}\left(\mathbf{x}\right) = \mathbf{P}\left(\mathbf{x}\right)\mathbf{W}\left(\mathbf{x}\right)\mathbf{P}\left(\mathbf{x}\right)^T \tag{1.58}$$

$$\mathbf{B}\left(\mathbf{x}\right) = \mathbf{P}\left(\mathbf{x}\right)\mathbf{W}\left(\mathbf{x}\right) \tag{1.59}$$

et

$$\mathbf{P}\left(\mathbf{x}\right) = \begin{bmatrix}\mathbf{p}\left(\mathbf{x}^{(1)} - \mathbf{x}\right) & \mathbf{p}\left(\mathbf{x}^{(2)} - \mathbf{x}\right) & \dots & \mathbf{p}\left(\mathbf{x}^{(n_s)} - \mathbf{x}\right)\end{bmatrix} \tag{1.60}$$

$$\mathbf{W}\left(\mathbf{x}\right) = \text{diag}\begin{bmatrix}w\left(\mathbf{x}^{(1)} - \mathbf{x}\right) & w\left(\mathbf{x}^{(2)} - \mathbf{x}\right) & \dots & w\left(\mathbf{x}^{(n_s)} - \mathbf{x}\right)\end{bmatrix} \tag{1.61}$$

$$\mathbf{y}_s = \begin{bmatrix}y\big(\mathbf{x}^{(1)}\big) & \dots & y\big(\mathbf{x}^{(n_s)}\big)\end{bmatrix}^T \tag{1.62}$$

$$\tag{1.63}$$

La fonction $w(\mathbf{x})$ joue un rôle prépondérant dans le comportement de l'approximation en assurant par exemple la continuité de l'approximation sur l'espace de conception considéré ou en rendant par exemple l'approximation interpolante. Quelques exemples de fonctions de pondération et les références associées peuvent être trouvés dans [NEALEN et DARMSTADT 2004].

Ce type de métamodèle a été employé dans le cadre de l'optimisation depuis le début des années 2000 pour l'optimisation fiabiliste [K. K. CHOI, YOUN et R.-J. YANG 2001], pour l'optimisation et les études stochastiques [TOROPOV et al. 2005] et pour obtenir une approximation locale de la fonction objectif [BREITKOPF et al. 2005].

4.5.4 Interpolation Cumulative

La méthode d'interpolation cumulative est inspirée des méthodes SPH (*Smooth Particle Hydrodynamics*) [GINGOLD et MONAGHAN 1977]. Elle repose sur la construction d'une approximation $\tilde{y}$ de la fonction y sous la forme d'une somme pondérée de fonction de forme ϕ_i de type exponentielle ou spline engendrant des approximations continues et dérivables sur le domaine d'étude. L'approximation s'écrit alors [J. RASMUSSEN 1998] :

$$\forall \mathbf{x}^{(0)} \in \mathcal{D},\ \tilde{y}\left(\mathbf{x}^{(0)}\right) = \frac{\sum_{i=1}^{n_s} \phi_i\left(\mathbf{x}^{(0)}, \mathbf{x}^{(i)}, k_i\right) y_i}{\sum_{i=1}^{n_s} \phi_i\left(\mathbf{x}^{(0)}, \mathbf{x}^{(i)}, k_i\right)} \tag{1.64}$$

où y_i est la valeur exacte de la fonction à approximer au point $\mathbf{x}^{(i)}$. Cette approximation n'est pas interpolante et un modèle d'interpolation cumulatif a été proposé par [SOULIER, RICHARD et al. 2003] :

$$\forall \mathbf{x}^{(0)} \in \mathcal{D},\ \tilde{y}\left(\mathbf{x}^{(0)}\right) = \frac{\sum_{i=1}^{n_s} \phi_i\left(\mathbf{x}^{(0)}\right) a_i}{\sum_{i=1}^{n_s} \phi_i\left(\mathbf{x}^{(0)}\right)} \text{ avec } \phi_i\left(\mathbf{x}^{(0)}\right) = e^{-k_i \|\mathbf{x}^{(0)} - \mathbf{x}^{(i)}\|^2} \tag{1.65}$$

Les fonctions ϕ_i sont donc des fonctions de forme de type gaussienne (d'autres fonctions sont envisageables [SOULIER et BOUCARD 2013]). Les coefficients a_i sont obtenus grâce à une méthode de collocation et par résolution du système d'équations linéaires suivant :

$$\mathbf{\Phi a} = \mathbf{y}_s \tag{1.66}$$

où

$$\mathbf{\Phi} = \begin{bmatrix} \dfrac{\phi_i\left(\mathbf{x}^{(1)}\right)}{\sum_i \phi_i\left(\mathbf{x}^{(1)}\right)} & \cdots & \dfrac{\phi_n\left(\mathbf{x}^{(1)}\right)}{\sum_i \phi_i\left(\mathbf{x}^{(1)}\right)} \\ \vdots & \ddots & \vdots \\ \dfrac{\phi_i\left(\mathbf{x}^{(n_s)}\right)}{\sum_i \phi_i\left(\mathbf{x}^{(n_s)}\right)} & \cdots & \dfrac{\phi_n\left(\mathbf{x}^{(n_s)}\right)}{\sum_i \phi_i\left(\mathbf{x}^{(n_s)}\right)} \end{bmatrix} \tag{1.67}$$

$$\mathbf{a} = \begin{bmatrix} a_1 & a_2 & \dots & a_{n_s} \end{bmatrix}^T \tag{1.68}$$

$$\mathbf{y}_s = \begin{bmatrix} y\left(\mathbf{x}^{(1)}\right) & \dots & y\left(\mathbf{x}^{(n_s)}\right) \end{bmatrix}^T \tag{1.69}$$

Les paramètres k_i permettent de contrôler la forme de l'approximation et sont déterminés par l'intermédiaire d'une méthode de validation croisée.

Le métamodèle considéré ici est continu et différentiable sur l'espace de conception et permet d'obtenir explicitement les gradients de la fonction approchée $\tilde{y}$.

Cette méthode d'interpolation cumulative a été employée dans un contexte d'optimisation d'assemblages (cf. par exemple [SOULIER et BOUCARD 2013]).

4.5.5 Réseaux de neurones

L'idée des réseaux de neurones a été proposé pour la première fois par McGulloch et Pitts en 1943 [MCCULLOCH et PITTS 1943]. Peu après, une règle d'apprentissage a été proposée [HEBB 1949] et le premier réseau perceptron [ROSENBLATT 1958] composé d'une couche de perception (les entrées) et d'une couche de décision (les sorties) a été introduit. Ce réseau est alors le premier système artificiel capable d'apprendre par expérience.

Dans un contexte mathématique, un neurone est une fonction algébrique non linéaire, paramétrée et à valeurs bornées. Un réseau de neurones est un assemblage par couches de neurones : la variable d'entrée des neurones d'une même couche est la combinaison linéaire des sorties des neurones de la couche précédente. Cet assemblage est construit de manière à définir une fonction de $\mathbb{R}^n$ dans $\mathbb{R}$. Pour définir un réseau de neurones, il faut donc choisir :

- le nombre de couches ;
- le nombre de neurones constituant chacune des couches ;
- les coefficients associés à chacune des combinaisons linéaires ;
- les paramètres définissant les neurones.

Ces coefficients représentent l'ensemble des degrés de liberté du réseau.

Les réseaux de neurones peuvent être classés en deux catégories :

- les neurones à « entrées paramétrées » pour lesquels les paramètres du réseau sont liés aux entrées du neurone (ce sont les coefficients associés aux différentes combinaisons linéaires) ;
- les neurones à « non-linéarités paramétrées », pour lesquels les paramètres sont attachés à la non-linéarité intrinsèque de la fonction.

Dans le cas de perceptrons multicouches (*multilayer perceptrons* - MP), qui sont des réseaux de neurones à « entrées paramétrées », la sortie $y_{j,k}$ du neurone j de la sous-couche k est de la forme

$$y_{j,k} = g_P\left(\psi_0 + \sum_{i=1}^{n_{k-1}} \psi_{u,j,k} y_{i,k-1}\right) \tag{1.70}$$

où n_{k-1} est le nombre de neurones de la sous-couche $k-1$, $y_{i,k-1}$ ($\forall i \in [\![1, n_{k-1}]\!]$) la sortie du neurone i de la sous-couche $k-1$ et $\psi_{i,j,k}$ les coefficients associés au neurone j.

La fonction g_P se nomme « *fonction d'activation* » du réseau. On la choisira généralement de la forme

$$g_P(x) = \frac{1}{1+\exp(-x)} \tag{1.71}$$

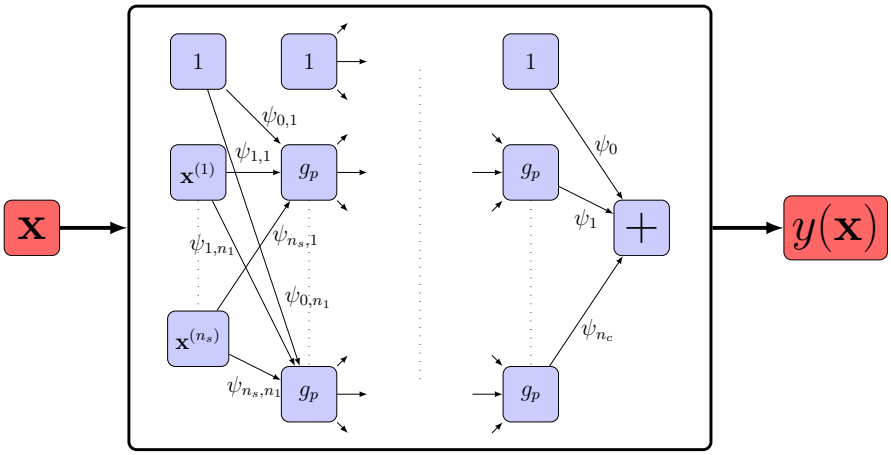


FIGURE 1.26 – Schéma de principe d'un perceptron multi-couche

On note généralement $\tilde{y}(\mathbf{x}, \boldsymbol{\psi})$ la sortie du réseau de neurones où $\boldsymbol{\psi}$ est le vecteur constitué de l'ensemble des coefficients définissant le réseau de neurones. Les coefficients sont solutions de la minimisation de l'erreur quadratique (MSE). Il s'agit de la phase d'apprentissage :

$$\boldsymbol{\psi} = \arg\min_{\boldsymbol{\psi}} \left(\text{MSE}(\boldsymbol{\psi})\right) = \arg\min_{\boldsymbol{\psi}} \left(\sum_{i=1}^{n_s} \left(\tilde{y}\left(\mathbf{x}^{(i)}\right) - y\left(\mathbf{x}^{(i)}\right) \right)^2 \right) \tag{1.72}$$

Ce problème est résolu par un algorithme d'optimisation (généralement un algorithme de descente dans la direction du gradient). En raison de la présence des couches successives, le calcul du gradient de l'erreur quadratique peut s'avérer particulièrement complexe. C'est pourquoi, généralement, on procède de manière récursive pour obtenir ce gradient.

La mise en oeuvre d'un modèle basé sur un réseau de neurones peut s'avérer extrêmement coûteuse en temps de calcul pour des problèmes de grande dimension. Les applications classiques employant les réseaux de neurones réalisent de la reconnaissance vocale et de la reconnaissance de caractères, les problèmes y sont très complexes et de grandes dimensions. Un certain nombre de travaux portent sur la parallélisation de cette démarche pour en diminuer le coût de [ROGERS et LAMARSH 1995]. Par ailleurs, on pourra se référer à [BISHOP 1995] pour plus de détails concernant les réseaux de neurones, leur mise en œuvre numérique et la détermination des paramètres.

Remarque 1.17 : Les réseaux de fonctions à base radiale (*Radial Basis Functions* - RBF) qui seront introduits dans la partie 2.2 page 156 peuvent être considérées comme des réseaux de neurones à « non-linéarités paramétrées ». Ils sont constitués d'une seule couche de fonctions à base radiales.

4.5.6 Régression à Vecteur de Support

La Régression à Vecteur de support est un métamodèle d'emploi relativement récent. Il est issu des Machines à Vecteur de Support (*Support Vector Machine* - SVM) développées dans les années 1990 [VAPNIK 1995] dans le cadre de la théorie de l'apprentissage. Initialement utilisées pour résoudre des problèmes de classification de données, elles ont été étendues à l'estimation de fonctions à partir d'un échantillon de sites et de réponses associées.

Relativement récente, cette méthode de régression a fait l'objet de nombreux développements et apparaît pour certaines personnes comme la méthode de régression la plus performante à l'heure actuelle. On pourra se référer à [GUNN 1998 ; SMOLA et SCHÖLKOPF 2004] pour des présentations détaillées du principe de la Régression à Vecteur de Support.

Dans la suite, le principe de ce métamodèle sera rappelé de manière succincte. En plus des références précédentes, il est possible de se référer à [FORRESTER, SÓBESTER et KEANE 2008 ; BOMPARD 2011] pour des exemples de mise en œuvre sur des fonctions uni-dimensionnelles et de dimensions plus élevées ainsi que pour des exemples pratiques de programmation de cette méthode.

Le principe de cette régression repose sur la construction de marges ε définissant une zone dans laquelle les réponses de la fonction à approcher n'affecteront pas la réponse prédite par le modèle de régression à vecteur support.

■ **Régression linéaire :** Dans un premier temps, on cherche à construire un estimateur linéaire du type :

$$\tilde{y}(\mathbf{x}) = \mu + \mathbf{w}^T\mathbf{x} \tag{1.73}$$

L'idée consiste alors à minimiser $\|\mathbf{w}\|^2$ [GUNN 1998] qui approche l'ensemble des points avec une précision ε donnée (on cherche donc à obtenir la fonction linéaire la plus plate). Le problème à résoudre s'écrit sous la forme du problème d'optimisation sous contraintes suivant :

$$\begin{aligned} &\min_{\mathbf{w},\mu,\boldsymbol{\xi}^+,\boldsymbol{\xi}^-} \frac{1}{2}\|\mathbf{w}\|^2 + C\frac{1}{n_s}\sum_{i=1}^{n_s}\left(\xi^{+(i)} + \xi^{-(i)}\right) \\ &\text{sous les contraintes } \forall i \in [\![1, n_s]\!] \begin{cases} y_i - \mathbf{w}^T\mathbf{x}^{(i)} - \mu \leq \varepsilon + \xi^{+(i)} \\ \mathbf{w}^T\mathbf{x}^{(i)} + \mu - y_i \leq \varepsilon + \xi^{-(i)} \\ \xi^{+(i)}, \xi^{-(i)} \geq 0 \end{cases} \end{aligned} \tag{1.74}$$

Le problème 1.74 propose un compromis entre le lissage de la fonction d'approximation et le respect de la tolérance ε en chaque point échantillonné. Ce compromis est régi par le paramètre $C \leq 0$. Cette approche est dite $\varepsilon - insensitive$. En effet, la formulation du problème revient à prendre en compte l'erreur sur les points échantillonnés par l'intermédiaire d'une fonction coût ne pénalisant pas les erreurs situées dans la zone définie par la marge ε (cf. [SMOLA et SCHÖLKOPF 2004]).

Le problème d'optimisation sous contraintes 1.74 est résolu en introduisant les multiplicateurs de Lagrange $\eta^{+(i)}$, $\eta^{-(i)}$, $\alpha^{+(i)}$ et $\alpha^{-(i)}$:

$$\begin{aligned}\mathcal{L}(\mathbf{w},\mu,\boldsymbol{\xi}^+,\boldsymbol{\xi}^-,\boldsymbol{\eta}^+,\boldsymbol{\eta}^-,\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-) =& \frac{1}{2}\|\mathbf{w}\|^2 + C\frac{1}{n_s}\sum_{i=1}^{n}\left(\xi^{+(i)}+\xi^{-(i)}\right)\\ &- \sum_{i=1}^{n_s}\left(\eta^{+(i)}\xi^{+(i)}+\eta^{-(i)}\xi^{-(i)}\right)\\ &- \sum_{i=1}^{n_s}\alpha^{+(i)}\left(\varepsilon+\xi^{+(i)}-y_i+\mathbf{w}^T\mathbf{x}^{(i)}+\mu\right)\\ &- \sum_{i=1}^{n_s}\alpha^{-(i)}\left(\varepsilon+\xi^{-(i)}+y_i-\mathbf{w}^T\mathbf{x}^{(i)}-\mu\right)\end{aligned} \tag{1.75}$$

Le problème revient alors à trouver $\mathbf{w}$, μ, $\boldsymbol{\xi}^+$ et $\boldsymbol{\xi}^-$ (variables primales) minimisant $\mathcal{L}$ et à trouver $\boldsymbol{\eta}^+$, $\boldsymbol{\eta}^-$, $\boldsymbol{\alpha}^+$, $\boldsymbol{\alpha}^-$ (variables duales où $\alpha^{\pm(i)} \geq 0$) maximisant $\mathcal{L}$. La recherche d'un point selle du Lagrangien $\mathcal{L}$ conduit aux égalités suivantes (pour les variables primales) :

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n_s}\left(\alpha^{+(i)}-\alpha^{-(i)}\right)\mathbf{x}^{(i)} = \mathbf{0} \tag{1.76}$$

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mu} = \sum_{i=1}^{n_s}\left(\alpha^{+(i)}-\alpha^{-(i)}\right) = 0 \tag{1.77}$$

$$\forall i \in [\![1,n_s]\!],\ \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\xi^{+(i)}} = \frac{C}{n_s} - \alpha^{+(i)} - \eta^{+(i)} = 0 \tag{1.78}$$

$$\forall i \in [\![1,n_s]\!],\ \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\xi^{-(i)}} = \frac{C}{n_s} - \alpha^{-(i)} - \eta^{-(i)} = 0 \tag{1.79}$$

de l'équation 1.76 on obtient

$$\mathbf{w} = \sum_{i=1}^{n_s}\left(\alpha^{+(i)}-\alpha^{-(i)}\right)\mathbf{x}^{(i)} \tag{1.80}$$

L'estimateur présenté par l'équation 1.73 s'écrit alors

$$\tilde{y}(\mathbf{x}) = \mu + \sum_{i=1}^{n_s}\left(\alpha^{+(i)}-\alpha^{-(i)}\right)\left({\mathbf{x}^{(i)}}^T\mathbf{x}\right) \tag{1.81}$$

■ **Régression non-linéaire :** Afin de pouvoir approximer des fonctions non-linéaires, l'approche classique consiste à employer une fonction de projection permettant de « simuler » le produit scalaire $\mathbf{x}\dot{\mathbf{x}}$ (cf. [SMOLA et SCHÖLKOPF 2004]). Il est alors possible d'employer une fonction noyau ϕ [15] (pour quelques exemples de fonctions, voir

15. La fonction ϕ doit être continue, symétrique et définie positive.

[GUNN 1998]) et réécrire l'estimateur proposé par l'équation 1.81 sous la forme :

$$\tilde{y}(\mathbf{x}) = \mu + \sum_{i=1}^{n_s} \left(\alpha^{+(i)} - \alpha^{-(i)}\right) \phi\left(\mathbf{x}^{(i)}, \mathbf{x}\right) \tag{1.82}$$

La détermination des coefficients $\alpha^{+(i)}$ et $\alpha^{-(i)}$ (que l'on nommera vecteurs de support) est réalisée en utilisant le problème dual basé sur le Lagrangien introduit par l'équation 1.75. En utilisant la fonction noyau et les résultats des équations 1.77, 1.78 et 1.79, le problème dual s'écrit :

$$\begin{aligned} \max_{\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-} & \left\{ \begin{array}{l} \dfrac{1}{2} \displaystyle\sum_{i=1}^{n_s} \sum_{j=1}^{n_s} \left(\alpha^{+(i)} - \alpha^{-(i)}\right) \left(\alpha^{+(j)} - \alpha^{-(j)}\right) \phi\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) \\ -\varepsilon \dfrac{1}{2} \displaystyle\sum_{i=1}^{n_s} \left(\alpha^{+(i)} + \alpha^{-(i)}\right) + \sum_{i=1}^{n_s} y_i \left(\alpha^{+(i)} - \alpha^{-(i)}\right) \end{array} \right. \\ \text{sous les contraintes} & \left\{ \begin{array}{l} \displaystyle\sum_{i=1}^{n_s} \left(\alpha^{+(i)} - \alpha^{-(i)}\right) = 0 \\ \forall i \in [\![1, n_s]\!],\ \left(\alpha^{+(i)}, \alpha^{-(i)}\right) \in [0, C/n]^2 \end{array} \right. \end{aligned} \tag{1.83}$$

En posant $\boldsymbol{\phi}$ la matrice de taille $n_s \times n_s$ contenant les termes $\phi\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)$, il est possible de réécrire le problème 1.83 sous une forme matricielle (et en transformant le problème de maximisation en problème de minimisation) :

$$\begin{aligned} & \min_{\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-} \frac{1}{2} \begin{bmatrix} \boldsymbol{\alpha}^+ \\ \boldsymbol{\alpha}^- \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\phi} & -\boldsymbol{\phi} \\ -\boldsymbol{\phi} & \boldsymbol{\phi} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}^+ \\ \boldsymbol{\alpha}^- \end{bmatrix} + \begin{bmatrix} \mathbf{1}^T \varepsilon - \mathbf{y}_s \\ \mathbf{1}^T \varepsilon + \mathbf{y}_s \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\alpha}^+ \\ \boldsymbol{\alpha}^- \end{bmatrix} \\ & \text{sous les contraintes} \left\{ \begin{array}{l} \mathbf{1}^T \begin{bmatrix} \boldsymbol{\alpha}^+ \\ \boldsymbol{\alpha}^- \end{bmatrix} = 0 \\ \forall i \in [\![1, n_s]\!],\ \left(\alpha^{+(i)}, \alpha^{-(i)}\right) \in [0, C/n]^2 \end{array} \right. \end{aligned} \tag{1.84}$$

Ce problème d'optimisation est un problème dit de programmation quadratique convexe que l'on peut résoudre à l'aide d'une méthode de points intérieurs [KARMARKAR 1984]. Les dimensions importantes des matrices et vecteurs de ce problème rendant cette méthode difficilement employable, d'autres stratégies peuvent être employées [BOMPARD 2011].

■ **Détermination des paramètres :** Les paramètres C, ε et les paramètres de la fonction noyau (s'il y en a) restent à déterminer. On se référera aux travaux de [BOMPARD 2011] qui présentent des références relatives aux stratégies de choix et de détermination de ces paramètres.

4.6 Validation

Une fois le métamodèle obtenu, il est nécessaire d'évaluer sa qualité. Pour cela, il est possible de se baser sur des critères nécessitant de nouvelles évaluations ou des approches basées uniquement sur les informations disponibles. On propose dans cette partie de présenter quelques critères d'évaluation de la qualité d'approximation d'un métamodèle.

4.6.1 MSE, RMSE

L'erreur des moindres carrés (*Mean Square Error*) est certainement l'indicateur le plus utilisé pour mesurer l'écart entre deux fonctions. Elle est employée dans le cadre de la régression polynomiale pour obtenir les coefficients de la décomposition polynomiale. Ces deux critères peuvent être utilisés pour mesurer l'écart entre la fonction approchée $\tilde{y}$ et la fonction réelle y en un ensemble de n_v points échantillonnés. Dans le cadre d'une méthode d'interpolation (comme c'est le cas pour la majorité des modèles de substitution déterministes), ces points doivent être différents de ceux utilisés pour construire le métamodèle.

$$\text{MSE} = \frac{1}{n_v}\sum_{i=1}^{n_v}\left(y\left(\mathbf{x}^{(i)}\right) - \tilde{y}\left(\mathbf{x}^{(i)}\right)\right)^2 \quad \text{RMSE} = \sqrt{\text{MSE}} \tag{1.85}$$

Au sens de ces critères, plus leur valeur est faible, plus le modèle de substitution est précis.

4.6.2 R^2

Le coefficient de corrélation linéaire de Bravais-Pearson traduit une dépendance linéaire entre deux variables en n_v point échantillonnés. Ce coefficient est couramment utilisé dans le cadre de l'emploi de modèles de régression.

$$R^2 = \left(\frac{\sigma_{xy}}{\sigma_x\sigma_y}\right)^2 \text{ où } \sigma_{xy} = \frac{1}{n_v}\sum_{i=1}^{n_v}\left(y\left(\mathbf{x}^{(i)}\right) - \overline{y}\right)\left(\tilde{y}\left(\mathbf{x}^{(i)}\right) - \overline{\tilde{y}}\right) \tag{1.86}$$

$$\sigma_x = \sqrt{\frac{1}{n_v}\sum_{i=1}^{n_v}\left(y(\mathbf{x}^{(i)}) - \overline{y}\right)^2} \quad \sigma_y = \sqrt{\frac{1}{n_v}\sum_{i=1}^{n_v}\left(\tilde{y}(\mathbf{x}^{(i)}) - \overline{\tilde{y}}\right)^2} \tag{1.87}$$

où $\overline{y}$ et $\overline{\tilde{y}}$ est les moyennes des réponses exactes et approchées prises aux n_v points.

Au sens de cette métrique : *plus la valeur R^2 est élevée plus le modèle réduit est précis* ($0 < R^2 \leq 1$).

4.6.3 Erreur relative moyenne absolue (RAAE)

$$\text{RAAE} = \frac{\sum_{i=1}^{n_s}\left|y\left(\mathbf{x}^{(i)}\right) - \tilde{y}\left(\mathbf{x}^{(i)}\right)\right|}{n_s\sigma} \tag{1.88}$$

où σ est l'écart type.

Au sens de cette métrique : *plus la valeur RAAE est faible plus le modèle réduit est précis*.

4.6.4 Erreur relative maximale absolue (RMAE)

$$\text{RMAE} = \frac{\max\limits_{i\in[\![1,n_v]\!]} \left|y\left(\mathbf{x}^{(i)}\right) - \tilde{y}\left(\mathbf{x}^{(i)}\right)\right|}{\sigma} \tag{1.89}$$

Cette métrique indique si une erreur importante est présente dans une zone de l'espace de conception. Des cas de figure peuvent se produire où le modèle réduit est très bon au sens de R^2 et RAAE mais mauvais au sens de RMAE. On préférera que l'erreur RMAE soit aussi petite que possible mais les deux premières erreurs R^2 et RAAE devront être prises en compte avant tout.

Au sens de cette métrique : *plus la valeur RAAE est faible plus le modèle réduit est précis*.

4.6.5 Quelques erreurs normées

Aux erreurs présentées précédemment, 3 nouvelles métriques ont été ajoutées. Celles-ci sont assez proches des précédentes à la différence qu'elles manipulent des données normées. Ainsi, on construit une erreur en chaque point :

$$\forall i \in [\![1, n_v]\!],\ e_i = \frac{\left(y\left(\mathbf{x}^{(i)}\right) - \tilde{y}\left(\mathbf{x}^{(i)}\right)\right)^2}{\sup\limits_{j\in[\![1,n_v]\!]} y\left(\mathbf{x}^{(i)}\right)^2} \tag{1.90}$$

On élabore ensuite les trois métriques. Plus elles sont faibles plus le métamodèle fournira une approximation globale de qualité :

$$\mathtt{Q}_1 = \sup_{i\in[\![1,n_v]\!]} e_i \quad \mathtt{Q}_2 = \sum_{i=1}^{n_v} e_i \quad \mathtt{Q}_3 = \frac{\mathtt{Q}_2}{n_v} \tag{1.91}$$

4.6.6 Erreurs basées sur l'emploi de la validation croisée

Les méthodes de validation croisée [STONE 1974 ; GEISSER 1975] sont très employées dans le domaine des statistiques depuis une quarantaine d'années pour choisir un modèle. On trouve classiquement 3 grands principes d'emploi de la validation croisée :

- le premier (*Hold-Out Validation*) consistant à diviser, avant la construction du modèle de substitution, les points échantillonnés et les réponses associées en deux ensembles : le premier est utilisé pour construire le métamodèle tandis que le second est employé pour vérifier l'approximation obtenue en comparant les réponses réelles disponibles et celles approchées par le métamodèle.
- l'approche de type *K-fold Cross-Validation* consistant à scinder les échantillons de points et les réponses associées en K groupes de même taille. Le processus de validation croisée consiste alors à élaborer K modèles de substitution construits chacun à partir de $K-1$ groupes d'échantillons. Le groupe restant est employé pour vérifier.

- l'approche de « biais en délaissant un site » (*Leave-One-Out Cross-Validation* - LOOCV) qui est un cas particulier de l'approche précédente où on construit n_s métamodèles construits chacun avec $n_s - 1$ points échantillonnés.

Ces approches fournissent chacune des informations statistiques du modèle considéré et permettent d'effectuer un choix lorsque l'on souhaite choisir un modèles parmi plusieurs. Dans le cas des modèles de substitution, l'approche LOOCV est assez couramment utilisée pour déterminer les paramètres internes du modèle minimisant par exemples l'erreur des moindres carrés (pour un modèle de substitution construit à partir de n_s points) :

$$\text{MSE}_{LOOCV} = \frac{1}{n_s}\sum_{i=1}^{n_s}\left(\tilde{y}_{-i}\left(\mathbf{x}^{(i)}\right) - y\left(\mathbf{x}^{(i)}\right)\right) \tag{1.92}$$

Cette approche est par exemple employée dans le cas de la détermination des paramètres des RBF (cf. partie 2.2.3 page 159 dans le chapitre 3).

4.7 Comparaisons de métamodèles

Les métamodèles présentés précédemment ne donnent qu'un bref aperçu des très nombreux modèles de substitution que l'on peut trouver dans la littérature. Le choix d'un modèle adapté à la problématique traitée s'avère complexe. Cependant, on trouve dans la littérature un nombre important de comparaisons de métamodèles. Malgré ces comparaisons, il est particulièrement complexe de faire un choix car elles se basent sur des exemples (analytiques ou physiques) différents qui présentent des fonctions à approximer plus ou moins non-linéaires, avec des types d'échantillonnage et de nombre d'échantillons différents et qui emploient des critères de comparaison différents d'une étude à l'autre. De plus, il n'existe pas réellement de métamodèles surpassant largement les autres.

Par ailleurs, il n'est pas présenté dans ce chapitre de mise en œuvre de métamodèles dans un contexte d'incertitudes mais le lecteur pourra se référer à [JIN, DU et W. CHEN 2003] pour quelques éléments de comparaisons concernant les métamodèles employés dans ce cadre.

Alors que certains travaux proposent d'employer simultanément plusieurs métamodèles [VIANA et HAFTKA 2008], un seul métamodèle sera utilisé dans cette thèse. Le tableau 1.2 suivant regroupe quelques articles de comparaison. Le contexte de ces comparaisons, le type d'échantillonnage, les types de métamodèles comparés et les conclusions des articles y sont regroupés.

Globalement, parmi les nombreux métamodèles et les nombreuses études comparatives, les métamodèles de krigeage et RBF apparaissent comme étant les métamodèles les plus polyvalents et les plus performants face à la majorité des situations. Ils nécessitent une attention particulière lors de leur construction (en particulier pour le krigeage) et leur emploi de manière générale. Comme nous le verrons dans la partie 3 page 160 du chapitre 3, ils possèdent également une extension permettant leur amélioration en incorporant les gradients de la fonction objectif.

RÉFÉRENCES	CONTEXTE(S)	MÉTAMODÈLES[a]	TIRAGE(S)	DIM.[b]	CONCLUSION(S)[c]
[T. SIMPSON et al. 1998]	Multidisciplinaire + optimisation	RP[1] KG (DACE)[2]	Tableau orthogonal	$25/3$ (25 KG)	**A** : krigeage saisi des perturbations supplémentaires **O** : Minima atteints très similaires
[GIUNTA, L. T. WATSON et KOEHLER 1998]	Analytique (2 fonctions)	RP[1] KG (DACE)[2] MOY RP/KG[3]	Manuel ($n_p = 1$) Plan D-optimal ($n_p = 5, 10$)	$3/1$ (201) $50/5$ (5^5) $100/10$ (10^3)	**A** : Les qualités d'approximations sont très similaires mais RP se détache sur un cas-test quasi-quadratique (et 1 cas-test avec $n_p = 10$ où le krigeage est inefficace).
[JIN, W. CHEN et T. SIMPSON 2000]	Analytique (13 fonctions) + Mécanique (1 cas-test)	RP[1] KG (DACE)[2] MARS RBF	Latin Hypercube Sampling	$n_p = 2, 3, 10, 14, 16$ 5 config. tirages (n_s) $n_v = 500, 1000 - 1200$	**A** : les RBF se distinguent des autres métamodèles (en termes de précision) mais restent assez proches du krigeage. Globalement lorsque le nombre d'échantillons est important, les métamodèles ont une qualité proche. RP reste préférable pour des fonctions faiblement non-linéaires et avec des tirages de petites dimensions. KRG reste coûteux.[4]
[VARADARAJAN, W. CHEN et PELKA 2000]	Multidisciplinaire	RP[1] ANN ANN/RP[5]	Central Composite Design	$273/8$ (121)	**A** : le modèle hybride ANN/RP propose une approximation de meilleure qualité que la RP. Par ailleurs, le modèle hybride assure une meilleure qualité d'approximation que le ANN. Pour des fonctions modérément et fortement non-linéaires, les choix se portent respectivement sur le ANN/RP et le ANN.
[PETER et al. 2007] [MARCELET 2008]	Mécanique + Optimisation	RP[6] ANN RBF KG[7]	Latin Hypercube Sampling	$-/8$ (441) enrichissement[8]	Plusieurs critères sont employés : qualité d'approximation minimum et localisation des minima locaux et globaux. **A** : RBF et KG sont les plus performants d'un point de vue qualité d'approximation. **O** : RBF et RP de degré 8 parviennent à localiser les minima locaux et globaux avec le moins de points échantillonnés. L'utilisation de KRG universel et d'un enrichissement adapté améliorent les performances de localisation.
[B.-S. KIM, LEE et D.-H. CHOI 2009]	Analytique (6 fonctions)	MLS[9] KG RBF SVR	Latin Hypercube Sampling	$n_s = 3n_p, 5n_p, 7n_p$ $n_p = 2, 4, 6, 8$ $n_v = 21^2, 15^4, 9^6, 6^8$	**A** : SVR et RBF fournissent les résultats les moins robustes par rapport au KG et MLS en termes de précision d'approximation. KG et MLS ont une précision similaire. Dans les cas à deux variables, KG fournit la meilleure approximation mais MLS fournit globalement la meilleure approximation.
[ZHAO et XUE 2010]	Analytique (6 fonctions)	RP RBF KG BNN	Latin Hypercube Sampling Aléatoire	$n_s = 3l(n_p + 1)(n_p + 2), l(n_p + 1)(n_p + 2)$ $l \in [0.5, 2]$ $n_p = 2, 10$ $n_v = 21^2, 15^4, 9^6, 6^8$	**A** : D'un point de vue précision, pour les problèmes de faible dimension, KG et RP semblent les plus appropriés tandis que pour des dimensions plus élevés, KG semble indiscutablement le plus approprié. Il reste par contre coûteux. D'un point de vue robustesse[10], le RP se détache largement des autres métamodèles. D'autres métriques sont employées dans ces travaux.

[a] RP : régression polynomiale, KG : krigeage, MOY : moyenne des réponses, MARS : *Multivariate Adpative Regression Splines*, RBF : *Radial Basis Functions*, ANN : *Artifical Neural Network*, BNN : *Bayesian Neural Network*, MLS : *Moving Least Square*, SVR : *Support Vector Regression*

[b] n_s/n_p (n_v x) où n_s est le nombre de sites, n_p le nombre de paramètres et n_v le nombre de points supplémentaires pour vérifier la qualité du modèle x.

[c] **A** : approximation, **O** : optimisation.

[1] Régression polynomiale à l'ordre 2.

[2] Terme de tendance constant et fonction de corrélation Gaussienne.

[3] Modèle mixte régression polynomiale d'ordre 2 et krigeage (proche d'un modèle de krigeage avec modèle de tendance polynomial).

[4] Dans notre cas, le coût de construction du métamodèle n'est pas un problème du fait du coût important nécessaire à l'obtention des réponses fournies par le modèle mécanique.

[5] Modèle mixte régression polynomiale d'ordre 2 réseau de neurones artificiels.

[6] Régression polynomiale de degrés 2, 4, 6 et 8.

[7] Krigeage Simple (modèle de tendance connu), krigeage Ordinaire (modèle de tendance constant inconnu) et krigeage Universel (modèle de tendance polynomial).

[8] Un enrichissement progressif des tirages est réalisé (le nombre de tirages minimum correspond au nombre de tirages nécessaires à déterminer tous les coefficients présents dans le métamodèle).

[9] Deux fonctions de pondération sont employées pour construire le métamodèle MLS.

[10] La robustesse est évaluée par la variabilité des critères d'erreurs pour plusieurs tirages ayant le même nombre d'échantillons.

TABLE 1.2 – Articles de comparaisons et conclusions

Remarque 1.18 : Des références complémentaires concernant des études de comparaison peuvent être trouvées dans [ZHAO et XUE 2010].

5 Bilan et cadre de l'étude proposée

En raison du coût de calcul lié aux multiples résolutions non-linéaires nécessaires à l'obtention d'un minimum local, cette thèse propose d'employer une approche multiniveaux de modèles pour mener à terme une optimisation d'assemblage. Parmi les méthodes d'optimisation introduites précédemment, une approche **multiniveaux de modèles hiérarchique** (cf. partie 2.4.3 page 37) formée de deux niveaux sera développée.

La résolution du problème mécanique d'assemblages sera réalisée en utilisant la stratégie multiparamétrique (cf. partie 3.2.3 page 59) qui permet de tirer partie des performances de la méthode LATIN (cf. partie 3.1 page 39). Les performances de cette stratégie de résolution et sa capacité à évaluer à moindre coût les gradients de la fonction objectif feront l'objet de la partie suivante (chapitre 2). À l'aide de cas-tests et en employant le code MATLAB capable de piloter le solveur, les spécificités de la stratégie seront présentées.

Dans un second temps et pour former le premier modèle de l'approche multiniveaux, les métamodèles seront développés (chapitre 3). En s'appuyant sur les études comparatives issues de la littérature et afin d'être capable de construire des métamodèles pouvant prendre en compte les gradients, les métamodèles de Krigeage et RBF seront rappelés en détail et leur version avec gradients explicitée. Les performances de ces métamodèles « @enrichis » seront présentées en employant des exemples analytiques et mécaniques.

Finalement, l'approche multiniveaux sera présentée en détail et appliquée sur des exemples d'optimisation d'assemblage complet (chapitre 4).

Chapitre 2

La stratégie multiparamétrique au service de l'optimisation d'assemblages

Ce chapitre présente la mise en œuvre de la stratégie multiparamétrique dans le cadre des calculs d'assemblages dédiés à l'optimisation. Après quelques mots sur la structure logicielle employée, les performances de la stratégie pour résoudre un ensemble de problèmes d'assemblages similaires seront présentées. Dans un second temps, la question du calcul des gradients sera abordée. L'état de l'art des méthodes existantes sera proposé puis une méthode basée sur les différences finies est adaptée à la stratégie multiparamétrique. Finalement, quelques performances de la méthode seront présentées sur un exemple simple.

Sommaire

1 Introduction 92
2 Mise en œuvre de la stratégie 92
2.1 Aspects pratiques des études paramétriques 92
2.2 Moyens de calcul 95
2.3 Présentation des performances 95
2.4 Bilan intermédiaire : 119
3 Calcul des gradients 119
3.1 Méthodes de calcul des gradients 119
3.2 Stratégie multiparamétrique et calcul des gradients 127
4 Illustrations des performances 130
5 Bilan sur l'emploi de la stratégie multiparamétrique pour l'optimisation 132

1 Introduction

Dans le contexte des études paramétriques ou de l'optimisation, il est nécessaire de résoudre un nombre important de problèmes très similaires. Ces problèmes sont très proches dans le sens où seuls quelques paramètres changent d'un calcul à l'autre. Dans le cadre des calculs d'assemblages tels qu'on les considère dans cette thèse, on s'intéresse à des paramètres présents aux interfaces entre les pièces (les sous-domaines). L'objectif ici est de chercher à mettre en œuvre une méthode permettant une réduction importante des temps de calculs d'une série de calculs similaires. Dans le cadre de l'emploi de la méthode LATIN, c'est naturellement que l'on se tourne vers la stratégie multiparamétrique pour résoudre ce problème. On propose dans ce chapitre de s'intéresser à la mise en œuvre de la stratégie multiparamétrique introduite dans la partie 3.2.3 page 59 du chapitre 1. Dans un premier temps, on se penchera sur des questions pratiques du contrôle et de l'exécution de multiples résolutions. Dans un second temps, les performances de la stratégie seront présentées en considérant plusieurs cas-tests 2D : différents types de paramètres pilotant le comportement des structures seront pris en compte. La question du choix d'une stratégie d'initialisation de l'algorithme sera étudiée. Dans le cadre de l'optimisation, il peut s'avérer utile de calculer les gradients de la fonction objectif considérée. Cette problématique fera l'objet de la troisième partie de ce chapitre. Dans un dernier temps, les performances de la stratégie multiparamétrique pour calculer un ensemble de réponses et de gradients seront présentées.

2 Mise en œuvre de la stratégie

2.1 Aspects pratiques des études paramétriques

L'emploi de la stratégie multiparamétrique et plus précisément du logiciel COFAST3D développé par Laurent CHAMPANEY [CHAMPANEY 1996] nécessite des aménagements particuliers pour pouvoir réaliser de manière aisée des études paramétriques ou des optimisations. Bien que le logiciel assure une mise en donnée aisée du problème, il ne permet pas de relancer successivement les différentes résolutions associées aux jeux de paramètres sans devoir dupliquer manuellement l'ensemble des procédures de prise en compte des paramètres et de post-traitement. Par ailleurs, COFAST3D et Cast3m sur lesquel il est basé n'incluent aucun outil permettant par exemple l'obtention des jeux de paramètres par échantillonnage de l'espace de variation des paramètres.

C'est pourquoi, dans le cadre de ces travaux, il a été retenu d'utiliser MATLAB pour

- générer les échantillonnages de l'espace de conception à l'aide d'outils disponibles dans les toolbox ou d'outils externes librement disponibles ;
- assurer l'exécution de COFAST3D pour chacun des jeux de paramètres échantillonnés ;

- ▶ récupérer les informations issues du post-traitement de COFAST3D (valeur de la fonction objectif, informations du solveur...).

2.1.1 Première solution

Initialement, Bruno SOULIER et Pierre-Alain BOUCARD avaient conçu un outil basé sur MATLAB permettant l'interfaçage de MATLAB et COFAST3D par l'intermédiaire de fichiers de script et de fichiers d'attente. Les fichiers scripts en langage Gibiane sont, pour chaque jeu de paramètres, modifiés afin d'y insérer les valeurs de paramètres. Ils sont ensuite copiés dans un fichier. Par l'intermédiaire d'un script `bash`[1] ou `Perl`[2], le fichier est alors lu par l'exécutable COFAST3D. Une fois cette lecture et l'exécution des commandes incluses réalisées, un fichier de fin est écrit. Pendant toute cette phase d'exécution, MATLAB se place en attente de ce fichier de fin qui, une fois détecté, l'efface et passe au jeu de paramètres suivant. Parmi les difficultés de cette approche, on relève la difficulté de débogguer le cas-test (en raison de conflits entre la sortie de COFAST3D et MATLAB, les scripts en langage Gibiane n'affichent aucun message en dehors des extractions d'informations volontaires), le peu de paramètres sur lesquels on peut agir depuis MATLAB (on ne peut agir sur les paramètres du calcul, seuls les paramètres de l'étude mécanique peuvent être modifiés) et les problématiques liées à la lecture et à l'écriture de fichiers intégralement dépendantes des performances et défaillances réseau et matériels.

2.1.2 Solution retenue, mise en place et développée

Afin d'améliorer les performances de ce processus et de remédier au problème d'usage des fichiers, une nouvelle version de cette interface MATLAB/COFAST3D a été conçue. Celle-ci se base sur l'emploi de **tubes nommés** (*named pipe* créé par la commande `mkfifo`). Ils s'apparentent à des fichiers mais sont en fait un mécanisme assurant le stockage des informations qu'ils contiennent en mémoire vive. La démarche employée consiste à écrire dans ces tubes les données qui sont ensuite lues par COFAST3D (principe FIFO soit *First In, First Out*). Dans la pratique et pour des raisons de simplicité de gestion, 3 tubes nommés sont utilisés : un assurant la communication de MATLAB vers COFAST3D (**IN**), un assurant le retour des informations de COFAST3D à MATLAB (**OUT**) et un permettant de connaître le statut de COFAST3D (**FLAG**), en particulier lorsqu'une exécution est arrivée à terme.

Par ailleurs, pour assurer un fonctionnement aisé de la stratégie multiparamétrique, le script COFAST3D nécessaire à l'exécution d'un cas est scindé en deux parties. Une partie est dédiée à l'initialisation et à la mise en données du problème (géométrie, maillage, spécification des interfaces, sous-domaines, comportements...) et est exécutée une seule fois, tandis que l'autre assure la modification des paramètres d'interface (obtenus par exemple par échantillonnage sous MATLAB), l'exécution du calcul associé à un jeu de paramètres

1. `https://www.gnu.org/software/bash/`
2. `http://www.perl.org`

et le post-traitement. Cette deuxième partie du script sera employée autant de fois qu'il y a de jeux de paramètres. Le diagramme de la figure 2.1 présente schématiquement le principe de communication MATLAB/COFAST3D par l'intermédiaire de l'interface basée sur les tubes nommés.

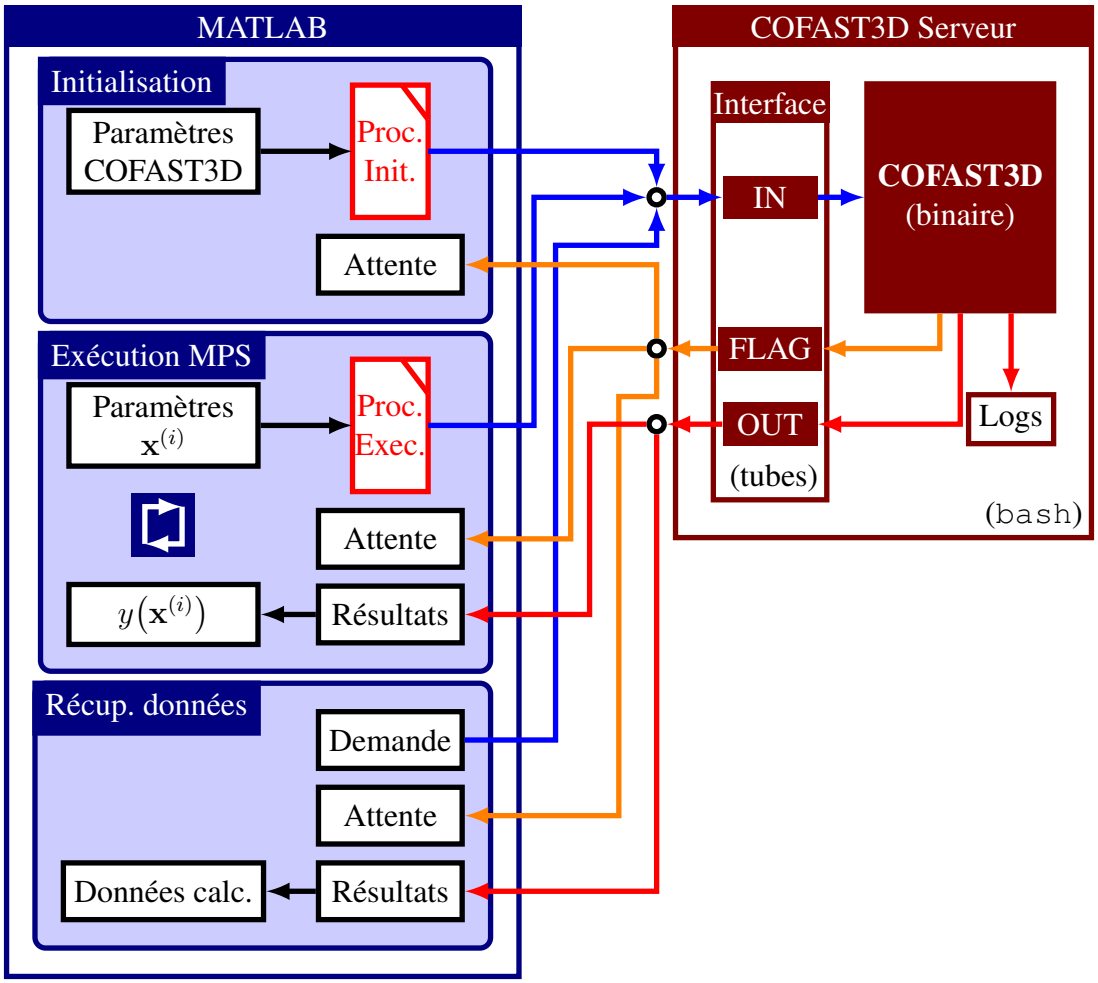


FIGURE 2.1 – Présentation synoptique du principe de pilotage de la stratégie multiparamétrique depuis MATLAB

Remarque 2.1 : Dans le cadre de la stratégie multiparamétrique avec contrôle de MATLAB telle qu'elle est présentée ci-dessus, COFAST3D peut être considéré comme un serveur (ou un *daemon*). C'est-à-dire qu'une fois lancé, le logiciel se place en phase d'attente d'instruction et restera actif tant qu'il ne subit pas d'interruption (suite à une erreur de script par exemple) ou qu'aucune instruction d'arrêt ne lui a été communiquée.

Remarque 2.2 : L'approche présentée ci-dessus comporte également un certain nombre de routines permettant le contrôle du déroulement du processus par l'intermédiaire de fichiers de logs et de débogage en cas de besoin.

Remarque 2.3 : Le dispositif d'interfaçage MATLAB/COFAST3D développé en interne au LMT-Cachan et l'ensemble des routines qui le compose ont été regroupés au sein de la toolbox COMET (***CO**ntrol **M**ultiparametric strat**E**gy **T**oolbox*). Ce code a été déployé dans une structure de versionnement à l'aide du logiciel `git` [a] qui permet un suivi extrêmement puissant des évolutions des sources, la gestion simplifiée d'évolutions parallèles (par exemple en cas de mise en place d'une version de test des routines) ainsi que le travail collaboratif au sein même de la toolbox et permettant une mise en commun aisée des travaux des différents participants. Résultat d'environ 3 ans et demi de développements, cette toolbox comporte actuellement 85 fichiers MATLAB, 7300 lignes de codes et environ 3000 lignes de commentaires et est le résultat de 800 interventions et actions dans les sources (écriture, déboggage). Elle intègre 30 cas-tests comportant de 1 à 12 paramètres portant sur 10 assemblages différents dont une partie sera présentée dans cette thèse.

a. `http://git-scm.com/`

2.2 Moyens de calcul

L'intégralité des exemples présentés dans ce manuscrit résulte d'exécutions sur le cluster du LMT-Cachan. Il comporte 70 noeuds de calcul soit 848 cœurs d'une puissance de calcul s'élevant à 11.3Tflops [3]. Ces noeuds sont équipés de processeurs Bi Xeon Quad Core 5450 (8 cœurs) et 5650 (12 cœurs), Bi-Xeon Octa Core (E5-2670, 16 cœurs) et Quadri-Xeon Octa Core (E7-8837, 32 cœurs). Ces processeurs sont épaulés de 32 Go à 256 Go de mémoire vive. De plus, la structure du cluster comporte environ 80 To d'espace disque de divers types (stockage, espace à accès rapide...). Toutes les machines fonctionnent sous une distribution Linux de type Debian et la gestion des exécutions sur le cluster est réalisée à l'aide du logiciel OpenPBS [4].

2.3 Présentation des performances

Dans cette partie, on s'intéresse à l'application de la stratégie multiparamétrique sur des cas d'assemblages simples. L'objectif est ici de présenter les performances de la méthode pour différents types de paramètres pris en compte. Afin de considérer des exemples pilotés quasiment intégralement par les phénomènes d'interface où les paramètres pris en compte interviennent, un ensemble d'exemples d'assemblages académiques comportant peu de sous-domaines et peu d'interfaces sera proposé. Il est important de noter que ces cas peuvent être considérés comme critiques vis-à-vis de la stratégie multiparamétrique : un changement même minime de la valeur du paramètre engendre une modification notoire du comportement de l'assemblage. Dans la pratique, sur un assemblage industriel, une modification de la valeur d'un paramètre d'interface peut dans certain cas n'avoir qu'une

3. *FLoating point Operations Per Second*
4. `http://www.adaptivecomputing.com/`

action très locale sans perturber l'ensemble de l'assemblage. Des exemples plus évolués seront bien entendu proposés dans les parties suivantes de ce manuscrit.

2.3.1 Mesure de performance

Dans le cadre de ces travaux de thèse, l'objectif principal est de mettre en œuvre une stratégie de résolution de problèmes d'assemblages assurant une réduction des temps de calcul. Afin d'évaluer cette réduction, un critère mesurant le gain en temps apporté par la stratégie multiparamétrique est proposé. Pour un ensemble de résolutions du problème d'assemblage associées à n_s jeux de paramètres $\mathbf{x}^{(i)}$, il évalue le rapport entre le temps de la stratégie multiparamétrique et de la méthode LATIN sans stratégie multiparamétrique. Il s'exprime de la manière suivante :

$$\text{Gain} = \frac{\text{Temps de calcul sans la stratégie multiparamétrique}}{\text{Temps de calcul avec la stratégie multiparamétrique}} \tag{2.1}$$

Malheureusement, dans la pratique, les n_s résolutions sont effectuées uniquement avec la stratégie multiparamétrique. Le calcul de ce critère nécessiterait $2 \times n_s$ résolutions (n_s résolutions avec la stratégie multiparamétrique et n_s résolutions sans celle-ci). Afin d'estimer, pour chaque calcul, le gain apporté par la stratégie, un critère de gain approché est en réalité employé. Pour n_s résolutions, il se calcule de la manière suivante :

$$\tilde{\text{Gain}} = \frac{n_s \times T_{ref}}{\text{Temps de calcul avec la stratégie multiparamétrique}} \tag{2.2}$$

où T_{ref} est un temps de référence choisi tel que $n_s \times T_{ref}$ soit une estimation du temps mis par l'approche sans stratégie multiparamétrique pour obtenir les n_s résolutions. En notant T_i le temps mis par la stratégie multiparamétrique pour obtenir la solution associée au jeu de paramètres $\mathbf{x}^{(i)}$, T_{ref} peut être choisi tel qu'il

- soit égal au temps du premier calcul réalisé, c'est-à-dire tel que

$$T_{ref} = T_1 \tag{2.3}$$

- soit égal au temps de calcul le plus long des n_s résolutions, c'est-à-dire tel que

$$T_{ref} = \max_{i \in [\![1, n_s]\!]} T_i \tag{2.4}$$

- soit égal au temps moyen des n_s résolutions, soit tel que

$$T_{ref} = \frac{1}{n_s} \sum_{i=1}^{n_s} T_i \tag{2.5}$$

Seuls les deux premiers temps de référence seront considérés. Généralement, le temps de la première résolution est le plus important et la stratégie multiparamétrique assure à chaque résolution suivante une réduction du temps. Dans quelques cas cependant, le comportement de l'assemblage peut entraîner une augmentation du temps de calcul pour une résolution. Dans ce cas, le second temps de référence est considéré.

Nous désignerons dans la suite par Gain_I et Gain_M respectivement le gain calculé en tenant compte du temps du premier calcul et le temps maximum des résolutions.

Ces critères de gains peuvent bien entendu être employés pour évaluer le gain de la stratégie multiparamétrique pour l'obtention d'**une seule** résolution associée à un jeu de paramètres $\mathbf{x}^{(i)}$. Ainsi ces critères se calculent de la manière suivante :

$$\forall i \in [\![1, n_s]\!],\ \text{Gain}_I\left(\mathbf{x}^{(i)}\right) = \frac{T_1}{T_i} \quad \text{Gain}_M\left(\mathbf{x}^{(i)}\right) = \frac{T^i_{max}}{T_i} \text{ où } T^i_{max} = \max_{j \in [\![1,i]\!]} T_j \tag{2.6}$$

Par ailleurs, le temps mesuré à chaque résolution correspond uniquement au temps des itérations de l'algorithme LATIN. Le temps de construction du problème et des opérateurs de la méthode ne sont pas pris en compte dans l'évaluation du gain. Ils ne sont cependant construits qu'une seule fois.

Remarque 2.4 : La version de COFAST3D utilisée lors de ces travaux de thèse est une version *mono-thread*. C'est-à-dire que lors du processus de résolution employant la méthode LATIN, seul un *thread* (ou tâche) sera utilisé (même si la machine comporte plusieurs cœurs). Un certain nombre d'opérateurs de Cast3m utilisés par COFAST3D lors de la mise en donnée de l'assemblage est parallélisé sous un aspect *multi-thread* (en employant autant de *threads* que de cœurs du processeur de la machine utilisée).

Remarque 2.5 : La mesure de temps employée est basée sur la procédure `TEMPS` de Cast3m qui mesure le temps CPU de chacune des opérations. Ce temps CPU est indépendant de la charge de la machine (si d'autres processus tournent sur celle-ci en parallèle de COFAST3D, ils n'ont pas d'impact sur le temps CPU du processus considéré) et est calculé en tenant compte de la priorité du processus, de l'architecture de la machine et du taux d'utilisation du processus (parmi les autres processus en cours d'exécution). Le temps CPU est un temps de référence indépendant de la machine sur laquelle le logiciel est employé.

2.3.2 Influence du type de paramètres

Dans cette partie, on s'intéresse aux performances associées à différents types de paramètres. Pour cela, on emploie 4 cas-tests 2D comportant chacun un paramètre variable. La particularité de ces cas-tests réside dans le fait que les paramètres pris en compte pilotent le comportement de la structure. Leur impact est donc très important et une modification, même mineure, de leur valeur se reflète quasi-intégralement sur la structure.

■ **Présentation des cas-tests :** Les cas-tests proposés ici sont des exemples relativement simples dont la géométrie sera réduite à des formes élémentaires (sous structures de forme rectangulaire). Ils comportent peu d'interfaces et ont des chargements simples et unidirectionnels (pour la plupart). Leur comportement est par contre non-linéaire mais limité à des zones précises de la géométrie (une ou deux interfaces avec frottement et/ou contact). La figure 2.2 présente la géométrie associée à chacun des cas-tests. Le tableau 2.1 montre les spécifications de ces cas-tests.

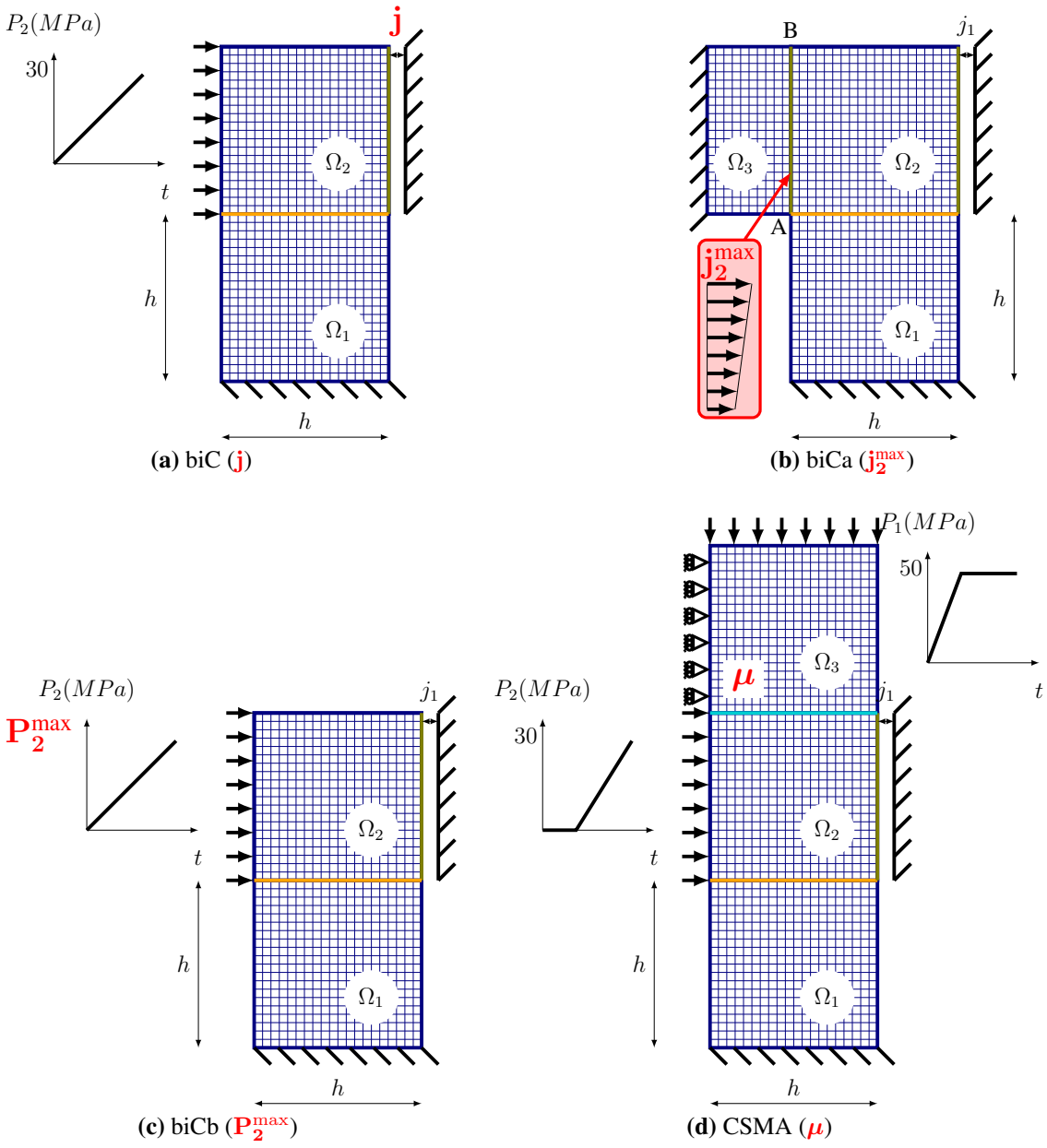


FIGURE 2.2 – Géométries des cas-tests (**en rouge et en gras**, les paramètres considérés)

CAS-TEST	biC	biCa	biCb	CSMA
Type d'éléments	QUA4			
Matériau	$E = 200\,\text{GPa}$ $\nu = 0.3$			
Type de simulation	Quasi-statique			
Nb. de pas de temps	20			
Paramètres fixes	$h = 50\,\text{mm}$ $j_1 = 0.04\,\text{mm}$			
Nb. SST / Nb. interfaces[a]	2/2	3/3	2/2	3/3
Interf. parfaites	$\Omega_1 - \Omega_2$	$\Omega_1 - \Omega_2$	$\Omega_1 - \Omega_2$	$\Omega_1 - \Omega_2$
Interf. frottement	–	–	–	$\Omega_2 - \Omega_3$
Interf. contact	$\Omega_2 -$ paroi	$\Omega_2 - \Omega_3$ et $\Omega_2 -$ paroi	$\Omega_2 -$ paroi	$\Omega_2 -$ paroi
DDL	1764	2226	1764	2646
Paramètres	jeu $\mathbf{j}$	précharge par champ de jeu $\mathbf{j_2^{max}}$	précharge par effort $\mathbf{P_2^{max}}$	coefficient de frottement μ
Plage de variation	0 à 0.24mm	0 à 0.1mm	100 à 1000N	0 à 0.8
Compléments		Champ de jeu appliqué progressivement de manière linéaire en temps et défini tel que en A $j = j_2^{\text{max}}/2$ et en B $j = j_2^{\text{max}}$	Effort variable appliqué progressivement de manière linéaire en temps de 0 à P_2^{max}.	Chargement en deux phases : précharge P_1 appliquée linéairement en temps de 0 à 50MPa puis fixe et charge P_2 appliquée linéairement en temps de 0 à 30MPa)

[a] hors interfaces avec les extérieurs (dédiées à l'application des conditions aux limites).

TABLE 2.1 – Spécifications des 4 cas-tests présentés dans la figure 2.2

Remarque 2.6 : Le cas-test nommé CSMA est un benchmark proposé par le « Groupe de travail CSMA sur les méthodes numériques de résolution des problèmes de contact unilatéral et de frottement ».

Remarque 2.7 : Le cas-test biCa propose d'imposer un champ de jeu positif. On notera que dans d'autres contextes, il est possible d'utiliser un champ de jeu négatif pour imposer un serrage.

■ **Réponses des assemblages :** Dans cette première étude, le critère d'arrêt (cf. partie 3.1.4 page 49 du chapitre 1) de l'algorithme LATIN sera fixé à $\mathbf{10^{-4}}$. On se propose pour chaque cas-test d'observer la valeur de la résultante de l'effort appliqué par la sous-structure Ω_2 sur la paroi. Par ailleurs, deux échantillonnages des paramètres sont réalisés pour chaque exemple : le premier consiste en un échantillonnage régulier comportant 20 jeux de paramètres et le second comporte 40 jeux de paramètres.

La figure 2.3 présente les efforts résultants sur la paroi. Les 4 cas-tests présentés ici comportent deux phases de comportement : une phase d'évolution quasi-linéaire en fonction du paramètre retenu et une phase constante correspondant au cas où la sous-structure Ω_2 n'est pas en contact avec la paroi. Les comportements de ces deux phases reflètent le comportement non-linéaire des assemblages. Par ailleurs, les deux échantillonnages fournissent des réponses concordantes.

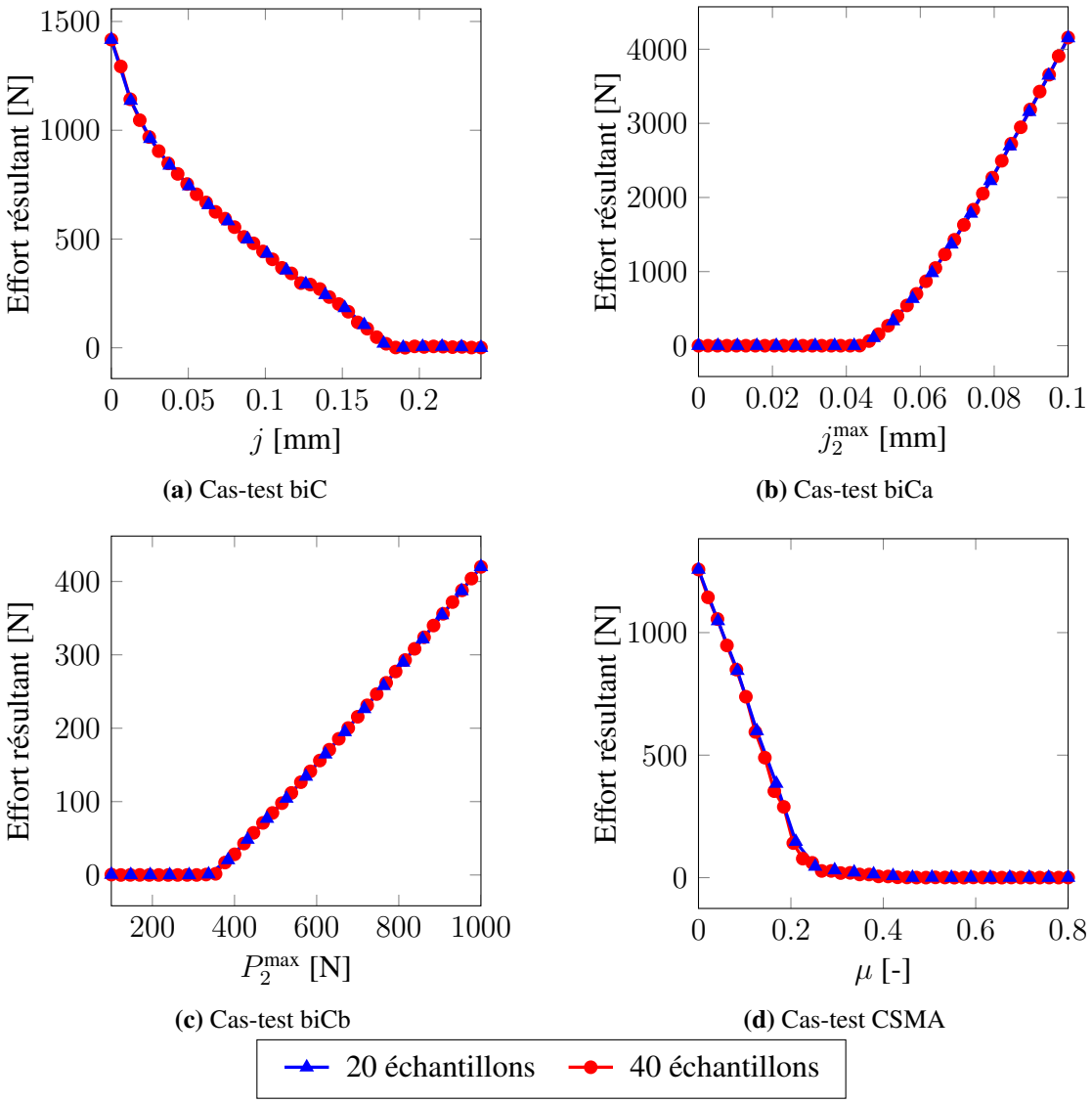
■ **Performances :** Afin d'illustrer les performances de la stratégie multiparamétrique sur les 4 cas-tests, les graphiques de la figure 2.4 présentent les temps de calcul et gains associés à l'obtention des réponses pour les deux échantillonnages considérés. Chaque graphique comporte, dans sa partie supérieure, les temps et gains pour l'échantillonnage à 20 jeux de paramètres et dans sa partie inférieure, les mêmes informations obtenues avec l'échantillonnage à 40 jeux de paramètres.

L'emploi de la stratégie multiparamétrique permet dans les 4 cas présentés ici de réduire les temps de calcul (cf. tableau 2.2). Cette réduction plus ou moins marquée selon les différents exemples est directement liée aux variations de la réponse de l'assemblage aux changements de valeur du paramètre. Seul le cas-test « biCb », où le paramètre est la valeur de l'effort appliqué sur la sous-structure Ω_2, ne permet pas de réduction aussi importante que les autres cas-tests : ce cas-test nécessite un nombre extrêmement important d'itérations de l'algorithme LATIN pour converger (avec un niveau d'erreur à 10^{-4}).

Sur les cas-tests « biC », « biCa » et « biCb », le critère de performance Gain_I peut être remis en cause. En effet, la première résolution n'étant pas la plus coûteuse en temps, elle ne permet pas de donner une estimation correcte du gain en temps de la stratégie multiparamétrique. Le critère Gain_M sera donc préférable dans la plupart des cas.

■ **Bilan :** Finalement, au vu des 4 exemples d'assemblages simples considérés ici, il est possible de distinguer un ordre d'importance de l'influence des paramètres vis-à-vis des performances de la stratégie multiparamétrique.

► Un paramètre de type effort aura une action importante sur le comportement de



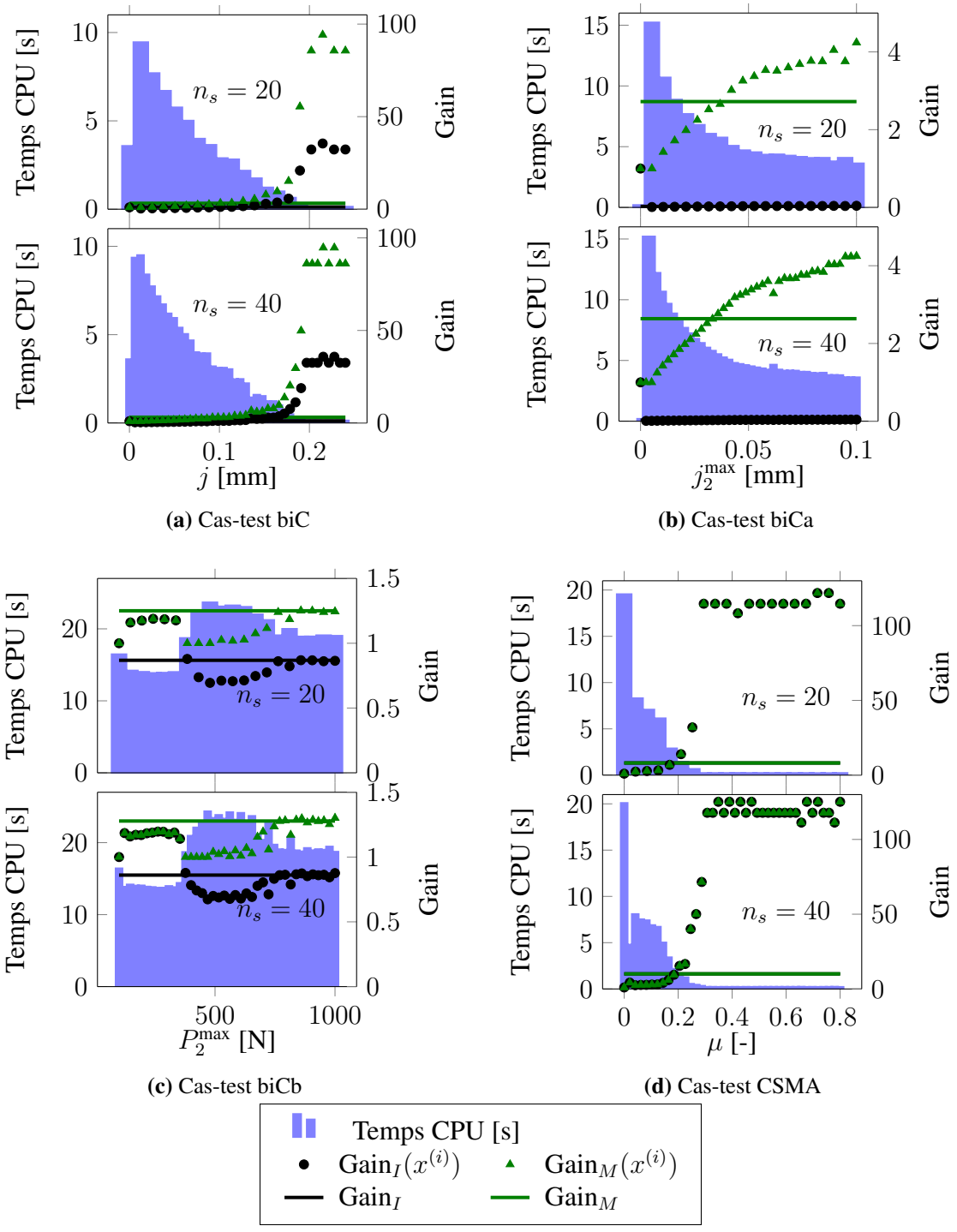
(a) Cas-test biC

(b) Cas-test biCa

(c) Cas-test biCb

(d) Cas-test CSMA

FIGURE 2.3 – Efforts résultants à la paroi



(a) Cas-test biC

(b) Cas-test biCa

(c) Cas-test biCb

(d) Cas-test CSMA

FIGURE 2.4 – Temps de calcul et gains pour les deux échantillonnages et pour les 4 cas-tests

CAS-TEST	biC		biCa		biCb		CSMA	
Type de paramètre	Jeu		Champs de jeu		Effort		Coef. frottement	
Nb. d'échantillons (n_s)	20	40	20	40	20	40	20	40
Temps CPU total [s]	58.51	121.2	111.53	229.88	377.45	759.93	47.87	79.43
Temps CPU moyen[a] [s]	2.93	3.03	5.58	5.75	18.87	19	2.39	1.99
Temps CPU mini.[b] [s]	0.1	0.1	0.15	0.15	13.79	13.67	0.16	0.16
Temps CPU maxi.[b] [s]	9.42	9.47	15.18	15.16	23.6	24.28	19.49	20.03
Nb. itérations total.	3 971	8 156	6 071	12 571	25 821	51 961	2 096	3 381
Nb. itérations moyen[a]	199	204	304	314	1 291	1 299	105	85
Nb. itérations mini.[b]	5	5	6	6	940	940	5	5
Nb. itérations maxi.[b]	645	645	835	835	1 610	1 670	876	876
Gain_I	1.21	1.18	0.03	0.03	0.87	0.86	8.14	10.09
Gain_M	**3.22**	**3.13**	**2.72**	**2.64**	**1.25**	**1.28**	**8.14**	**10.09**

[a] Temps CPU et nombre d'itérations moyen par calcul.
[b] Temps CPU et nombre d'itérations minimum et maximum des n_s résolutions.

TABLE 2.2 – Quelques caractéristiques des calculs sur les 4 assemblages (critère d'arrêt LATIN $\eta = 10^{-4}$)

l'algorithme et le changement de sa valeur nécessitera, pour obtenir la solution au nouveau problème, un nombre important d'itérations.

- Un paramètre de type coefficient de frottement aura un effet moins important et la stratégie multiparamétrique permettra d'importantes réductions des temps de calcul dans le cas où l'on souhaite réaliser de nouvelles résolutions associées à des coefficients de frottement différents (gain d'environ 10).

- Un paramètre de type jeu (jeu uniforme ou non) aura un impact plus modéré sur le processus itératif mais la stratégie multiparamétrique permettra quand même des réductions des temps de calcul non négligeables (avec un gain d'environ 3).

Par ailleurs, lorsque le comportement de la structure évolue peu entre deux jeux de paramètres (ce qui est le cas lorsqu'il n'y a pas contact entre la sous-structure Ω_2 et la paroi), les performances de la stratégie multiparamétrique peuvent être très importantes : le gain peut atteindre des valeurs proches de 100 (voir par exemple le cas-test biC).

Remarque 2.8 : Comme cela a déjà été précisé au début de cette étude, les cas-tests d'assemblages considérés ici présentent des paramètres jouant un rôle prépondérant sur le comportement et la réponse de l'assemblage. C'est pourquoi, dans certains cas, les réductions des temps de calcul peuvent sembler faibles. Dans le cas d'un assemblage réel, les paramètres auront une influence beaucoup plus localisée dans l'assemblage et les performances associées à ce type de problèmes s'en trouveront améliorées.

2.3.3 Influence du critère d'arrêt

Comme nous l'avons vu dans la partie 3.1.4 du chapitre 1 page 49, la convergence de l'algorithme LATIN peut être contrôlée à l'aide d'un critère d'arrêt basé sur la vérification des relations de comportement des interfaces. La valeur de ce critère d'arrêt (fixé à 10^{-4} dans la partie précédente) joue un rôle prépondérant sur le nombre d'itérations de l'algorithme lors d'une résolution. On propose dans cette partie d'étudier l'influence du critère d'arrêt sur les performances de la stratégie multiparamétrique. Pour cela, les 4 cas-tests introduits dans la partie précédente seront réutilisés. On se limitera à utiliser l'échantillonnage régulier comportant 20 sites.

■ **Influence sur la réponse :** Dans un premier temps et pour chaque cas-test, les valeurs des fonctions objectifs aux différents points échantillonnés sont observées pour différentes valeurs du critère d'arrêt de l'algorithme LATIN. Ces résultats sont visibles pour 9 valeurs du critère d'arrêt dans la figure 2.5.

Comme on peut l'observer, la valeur du critère d'arrêt de l'algorithme LATIN a un impact extrêmement important sur la réponse de l'assemblage. A l'exception du cas-test « CSMA », la valeur du critère fixée dans la partie précédente (10^{-4}) permet d'obtenir une précision satisfaisante de la réponse. La figure 2.6 présente l'erreur en chaque point échantillonné calculée en employant la formule suivante :

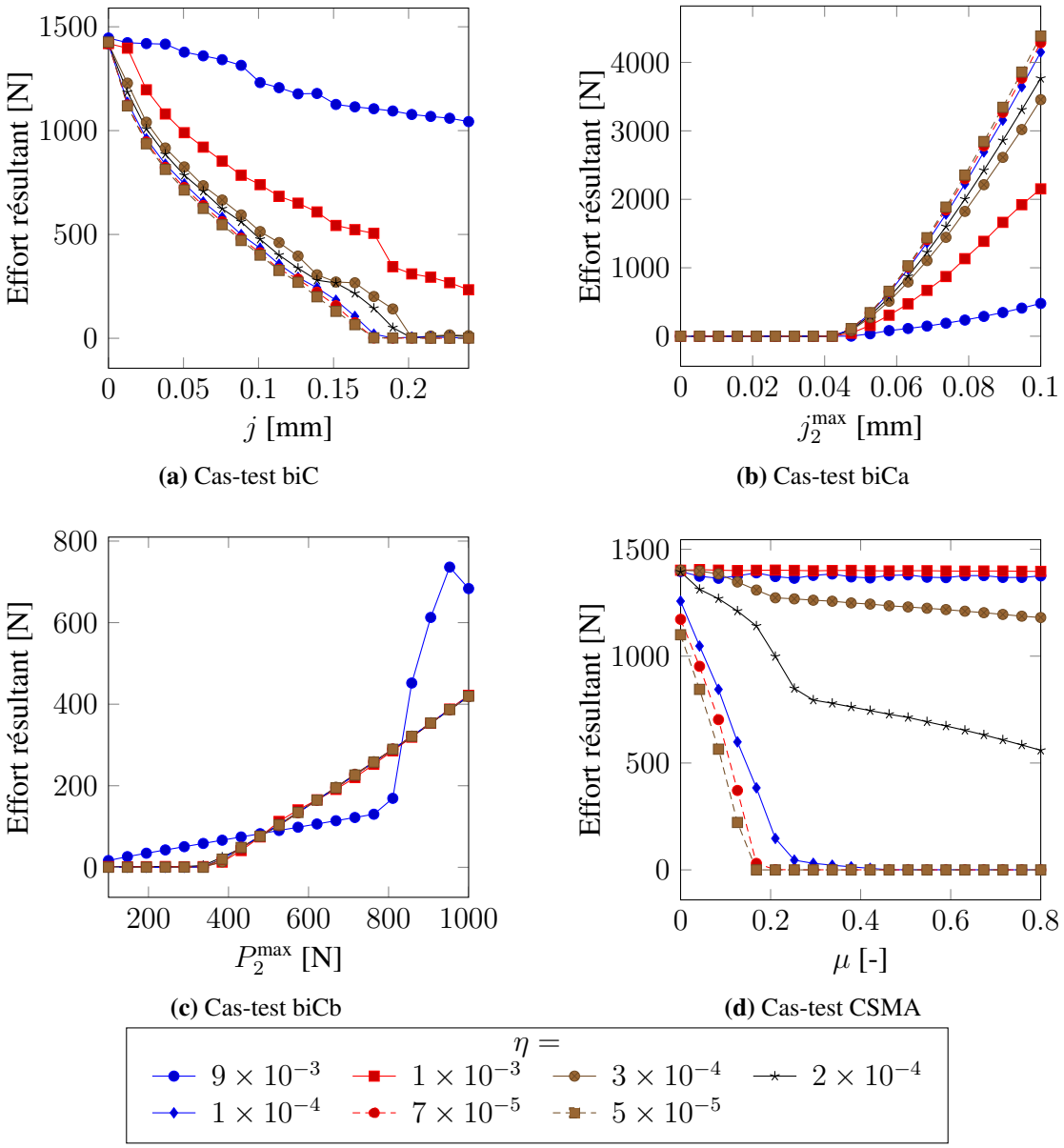
$$\forall i \in [\![1, n_s]\!],\ E_\eta\left(\mathbf{x}^{(i)}\right) = \left|F_\eta\left(\mathbf{x}^{(i)}\right) - F_{\eta=5\times10^{-5}}\left(\mathbf{x}^{(i)}\right)\right| \tag{2.7}$$

$E_\eta\left(\mathbf{x}^{(i)}\right)$ est donc l'écart calculé au site $\mathbf{x}^{(i)}$ pour un critère d'arrêt fixé à η en tenant compte de la valeur de la réponse obtenue avec un critère fixé à η et celle obtenue avec une convergence importante où le critère est fixé à 5×10^{-5}. Il n'a pas été possible d'employer une erreur relative en raison des valeurs prises par l'effort lorsqu'il n'est pas en contact avec la paroi (effort nul).

On observe sur ces courbes un écart qui peut, dans certains cas, paraître important. Il est bien entendu à relativiser avec la valeur prise par la fonction dans ce cas. Par exemple pour le cas-test CSMA, on observe pour un niveau d'erreur $\eta = 10^{-4}$ une erreur approchant les 100 N. Or, dans cette phase, l'effort transmis par la sous-structure 2 est de l'ordre de 4000 N. Ce qui revient à une erreur relative acceptable de l'ordre de 2.5%.

Globalement, sur l'ensemble des cas-tests, il est possible de faire le même constat. Dans les zones où l'effort est nul ou proche de zéro, à l'exception du cas-test CSMA, un critère LATIN de l'ordre de $\eta = 10^{-3}$ pourra fournir une réponse de qualité proche d'une réponse hautement convergée. Par contre, dans les phases où l'effort à la paroi n'est pas nul, il est important de considérer un critère d'erreur de l'ordre de $\eta = 10^{-4}$. Il serait bien entendu idéal de choisir un critère d'erreur très faible mais les performances de la méthode risqueraient d'être clairement déteriorées.

■ **Influence sur les performances :** Le choix d'une valeur de critère d'arrêt de l'algorithme LATIN fait apparaître un réel dilemme précision-performance. En effet, choisir un critère d'arrêt faible engendre un accroissement du nombre d'itérations et par



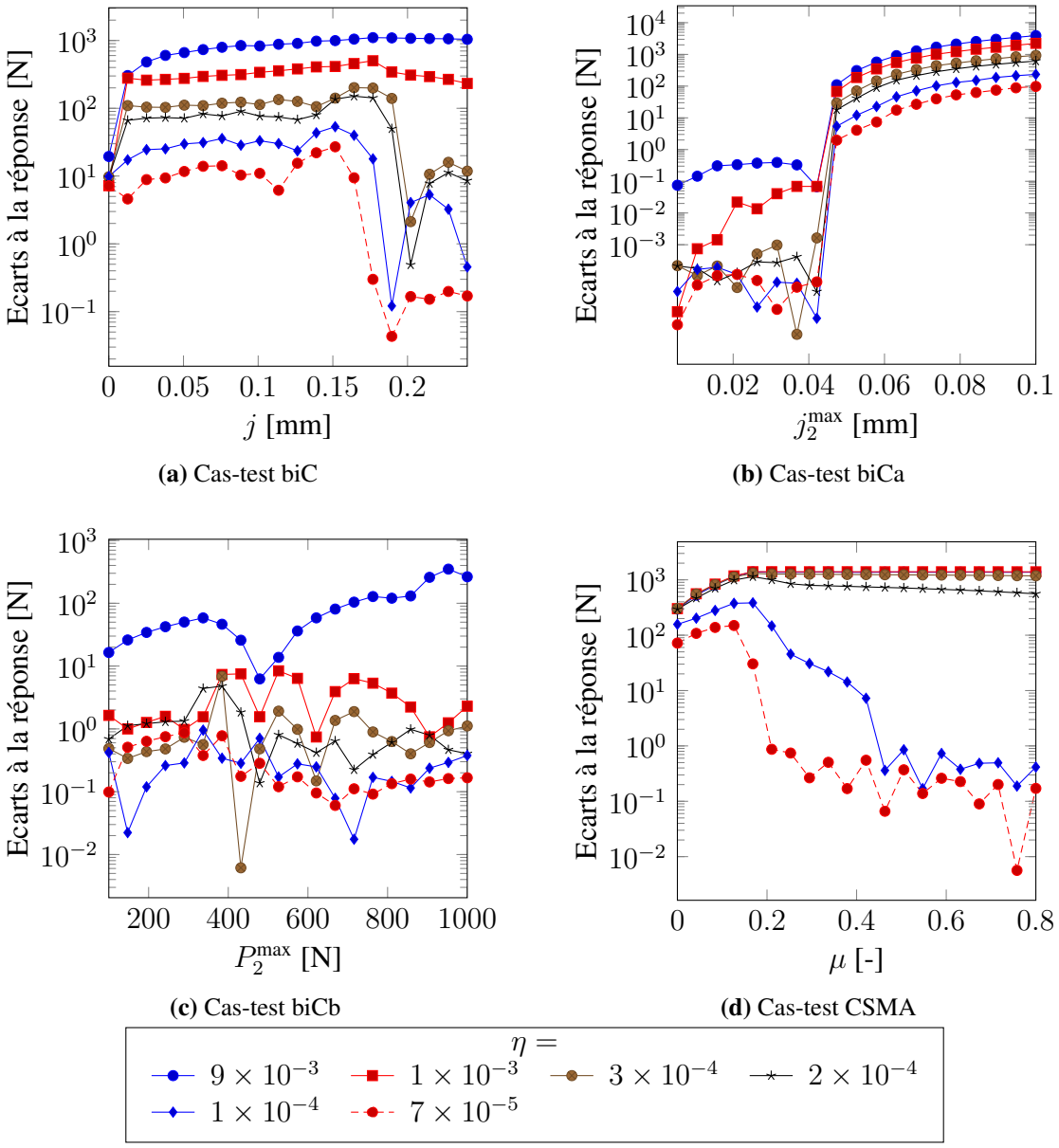
(a) Cas-test biC

(b) Cas-test biCa

(c) Cas-test biCb

(d) Cas-test CSMA

FIGURE 2.5 – Efforts résultants à la paroi



(a) Cas-test biC

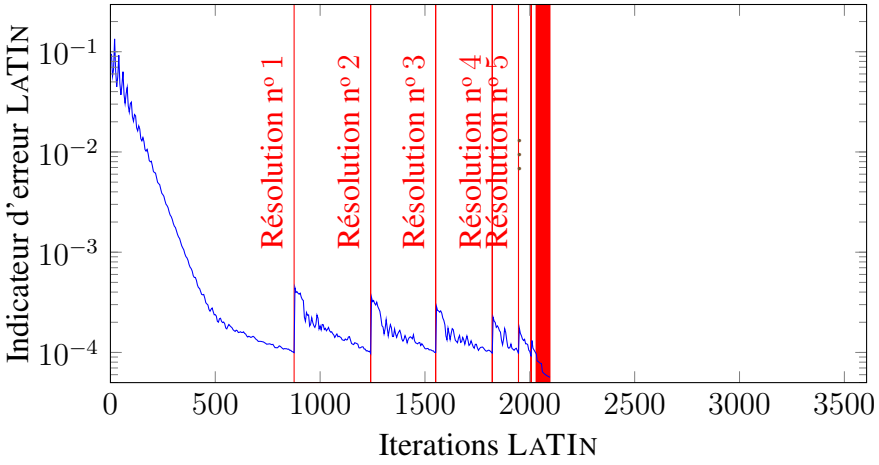
(b) Cas-test biCa

(c) Cas-test biCb

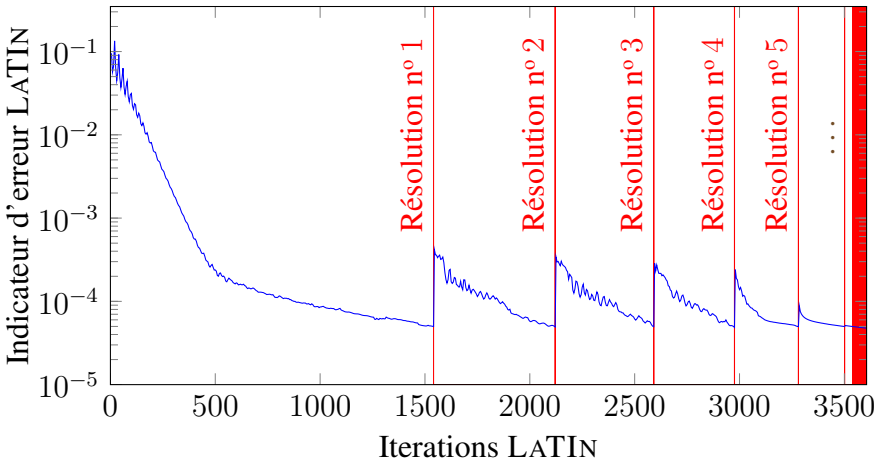
(d) Cas-test CSMA

FIGURE 2.6 – Ecart sur la résultante

conséquent du temps CPU associé à une résolution. La figure 2.7 illustre ce phénomène : dans le cadre de l'emploi de la stratégie d'optimisation pour le cas-test « CSMA » et pour un échantillonnage comportant 20 points, les 2 courbes présentent chacune l'évolution de l'indicateur d'erreur lors des 20 résolutions réalisées avec 2 critères d'arrêt différents (10^{-4} et 5×10^{-5}).



(a) 2096 itérations à $\eta = 10^{-4}$



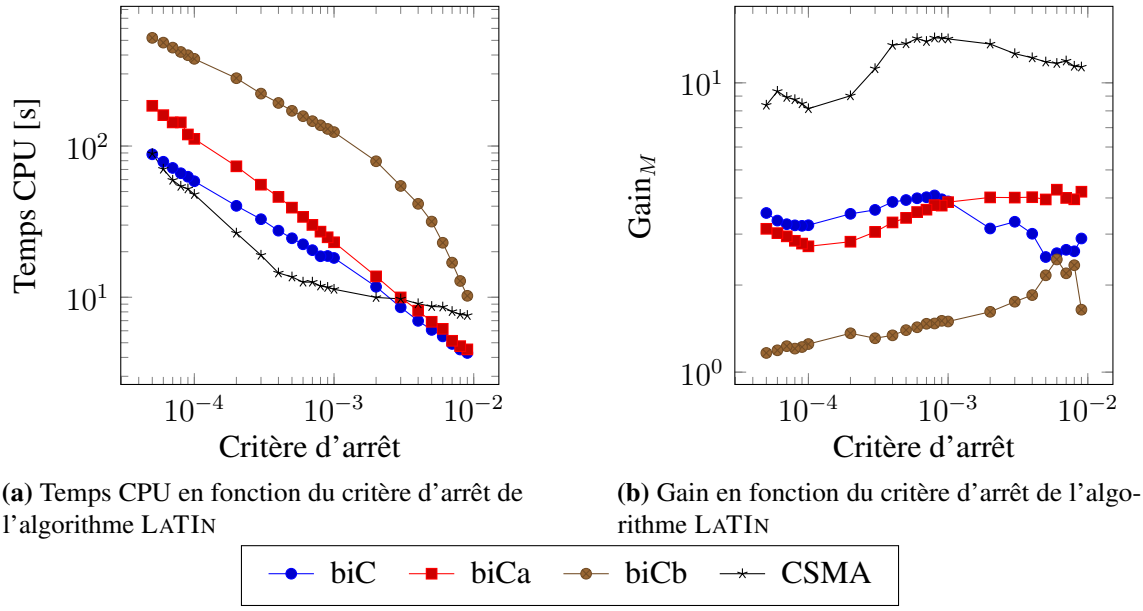
(b) 3606 itérations à $\eta = 5 \times 10^{-5}$

FIGURE 2.7 – Courbes de convergence de la stratégie multiparamétrique du cas-test « CSMA » pour 20 jeux de paramètres et 2 critères d'arrêt de l'algorithme LATIN

Sur ces figures apparaissent les fortes variations de l'indicateur d'erreur LATIN liées aux changements de valeurs des jeux de paramètres et marquant le début d'une nouvelle résolution. Par ailleurs, grâce à cet exemple, on constate une forte différence du point de vue du nombre total d'itérations. En effet, dans le cas où le critère d'arrêt est fixé à 10^{-4}, les 20 résolutions sont réalisées en 2096 itérations et un temps CPU de 47.9 s, tandis que lorsque le critère d'arrêt est fixé à 5×10^{-5}, elles sont obtenues en 3606 itérations et 89.2 s. De plus, pour les deux critères d'arrêt retenus, on observe qu'à partir du moment où la

sous-structure Ω_2 n'est plus en contact avec la paroi (la résultante de l'effort à la paroi est nulle, cf. figure 2.3d page 101 ou figure 2.5d page 105), les changements de paramètres ont un impact négligeable sur le nombre d'itérations pour obtenir la solution associée pour les deux critères d'arrêt.

En complément, pour un nombre plus important de critères d'arrêt et pour les 4 exemples d'assemblages, on peut également évaluer le temps CPU nécessaire aux évaluations. Ces résultats sont présentés par la figure 2.8 sous la forme de graphiques en échelle logarithmique. On observe sur les 4 cas-tests que le temps de calcul dépend du critère d'arrêt sous la forme d'une fonction puissance. Ce constat se vérifie particulièrement pour les cas-tests « biC » et « biCa ». Le gain, quant à lui, dépend peu du critère d'arrêt pour les 4 exemples. Les constats liés à l'influence du type de paramètres (cf. partie 2.3.2 page 100) se confirment également dans cette situation : les gains les plus élevés sont obtenus en considérant comme paramètre un coefficient de frottement.



(a) Temps CPU en fonction du critère d'arrêt de l'algorithme LATIN

(b) Gain en fonction du critère d'arrêt de l'algorithme LATIN

FIGURE 2.8 – Temps CPU et gain de la stratégie multiparamétrique en fonction du critère d'arrêt pour les 4 exemples d'assemblages

■ **Bilan :** Suite à cette étude, le choix du critère d'arrêt de la méthode LATIN est prépondérant à la fois sur la qualité de la réponse obtenue et sur les performances de la stratégie multiparamétrique. Dans l'objectif de trouver un compromis précision/performances, on propose de fixer le critère d'arrêt à 10^{-4}. Cette valeur pourra bien entendu nécessiter une étude complémentaire pour assurer un choix fin dans le cas où l'on traiterait par exemple un problème 3D.

2.3.4 Problématique de l'initialisation de la stratégie mutliparamétrique

Comme cela a été introduit dans la partie 2.3.4 du chapitre 1 page 109, il est nécessaire avant chaque résolution employant la stratégie multiparamétrique d'initialiser les grandeurs d'interface de la méthode LATIN avec des grandeurs obtenues lors d'une résolution précédente. Dans la partie précédente, où les cas-tests ne présentaient qu'un seul paramètre, l'algorithme LATIN était initialisé avec les grandeurs issues de la résolution précédente. Dans un contexte où l'on peut avoir affaire à des assemblages comportant plusieurs paramètres, il est important de chercher la meilleure résolution précédente $\mathbf{s}_i$ ($i \in [\![1, k-1]\!]$) permettant d'initialiser l'algorithme afin d'obtenir la solution $\mathbf{s}_k$ ($k \in [\![1, n_s]\!]$). Cette solution devra permettre de minimiser le temps de calcul de la solution $\mathbf{s}_k$. Plus globalement, il est pertinent de chercher, pour chaque résolution, une solution d'initialisation permettant de minimiser le coût global des n_s résolutions réalisées.

Dans cette partie, on s'intéresse à la problématique de choix d'une stratégie de réinitialisation. Pour cela, il est possible d'identifier deux contextes distincts dans lesquels ce choix peut être envisagé :

- celui où l'on souhaite seulement obtenir la solution $\mathbf{s}_k$ associée au jeu de paramètres $\mathbf{x}^{(k)}$ avec $k-1$ solutions $\mathbf{s}_i$ ($i \in [\![1, k-1]\!]$) disponibles pour initialiser l'algorithme LATIN sans connaissance *a priori* d'autres jeux de paramètres $\mathbf{x}^{(j)}$ pour lesquels les solutions $\mathbf{s}_j$ ($j > k$) seront par la suite évaluées.

- celui où l'on connaît initialement l'ensemble des n_s jeux de paramètres $\mathbf{x}^{(k)}$ pour lesquels on souhaite obtenir la réponse du modèle d'assemblage.

La seconde situation est celle que l'on rencontre par exemple lorsque l'on construit un métamodèle et que l'ensemble des jeux de paramètres sont échantillonnés en utilisant une méthode de type plan d'expérience numérique (*Latin Hypercube Sampling* par exemple, voir la partie 4.4.2 du chapitre 1 page 69). La première situation est rencontrée par exemple lors d'une phase d'enrichissement d'un métamodèle où les nouveaux jeux de paramètres sont obtenus au fur et à mesure du processus ou lors d'un processus d'optimisation où l'optimiseur déterminera les jeux de paramètres.

On s'intéresse ici principalement à la seconde situation avec pour objectif de tirer profit de la connaissance *a priori* des jeux de paramètres. Ainsi, dans l'objectif de minimiser les coûts de calcul en employant au mieux la stratégie multiparamétrique, on peut identifier 3 facteurs pouvant influencer le temps de calcul global nécessaire à l'obtention des n_s solutions :

- la stratégie de choix de la solution d'initialisation, basée sur l'emploi d'une distance dans l'espace des paramètres par exemple ;

- le choix du premier des n_s jeux de paramètres pour lequel la solution sera calculée ;

- l'ordre dans lequel les jeux de paramètres sont triés.

Afin d'élaborer une stratégie d'initialisation pertinente permettant de minimiser le temps CPU en employant la stratégie multiparamétrique, les 3 facteurs influents seront par la suite étudiés sur un assemblage simple qui comporte 2 paramètres.

2.3.5 Stratégies d'initialisation

Le choix de la meilleure stratégie de réinitialisation, c'est-à-dire celle permettant d'améliorer les performances d'un point de vue du temps de calcul, est complexe. En effet, l'idée repose sur une connaissance *a priori* de l'interdépendance de deux solutions mécaniques associées à deux jeux de paramètres distincts : il paraît plus judicieux d'initialiser l'algorithme LATIN avec une solution précédemment convergée reflétant un comportement proche de celui qui est attendu. On peut citer un exemple simple où, pour une solution $\mathbf{s}_i$ à un problème d'assemblage faisant intervenir un comportement de contact entre deux sous-structures, il sera préférable d'initialiser l'algorithme LATIN avec une solution précédemment convergée faisant intervenir ce même phénomène. Le principal défaut de ce raisonnement est qu'il s'appuie sur le fait que lorsque l'on cherche une solution $\mathbf{s}_i$ associée à un nouveau jeu de paramètres $\mathbf{x}^{(i)}$, le comportement de l'assemblage est connu. Malheureusement, dans la pratique, ce comportement est inconnu et cette problématique visant à trouver les phénomènes mécaniques similaires n'est par conséquent pas viable.

Néanmoins, il est possible de réduire cette problématique de recherche de solution proche en employant différentes stratégies [SOULIER et BOUCARD 2009] :

- la première (que l'on désignera sous le nom `init`) consiste simplement à réinitialiser l'algorithme LATIN avec la solution $\mathbf{s}_1$ obtenue en premier.

- dans un second temps, il est possible d'initialiser l'algorithme en employant la solution associée au jeu de paramètres précédent (dans l'ordre des résolutions). Ainsi,

$$\mathbf{s}_i{}^{(0)} = \mathbf{s}_{i-1} \tag{2.8}$$

  Cette méthode sera désignée sous le terme `auto`.

- il est également possible de chercher la solution précédente la plus proche dans l'espace des paramètres. L'équation 2.9 présente ce principe pour une norme-p où, pour initialiser l'algorithme LATIN pour obtenir la solution $\mathbf{s}_i$, on cherche le jeu de paramètres le plus proche de $\mathbf{x}^{(i)}$.

$$\mathbf{s}_i{}^{(0)} = \mathbf{s}(\mathbf{x}^{\text{init}}) \text{ tel que } \mathbf{x}^{\text{init}} = \underset{j \in [\![1, i-1]\!]}{\arg\min} \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_p \tag{2.9}$$

  Cette stratégie se rapproche le plus de la méthode consistant à trouver la solution associée aux phénomènes mécaniques dans la mesure où une faible variation des paramètres peut avoir un impact faible sur le comportement de l'assemblage. Cependant, des situations où de faibles variations de paramètres peuvent engendrer des changements radicaux du comportement (par exemple avec un contact présent dans une solution mais pas dans l'autre). Il est possible de concevoir plusieurs mesures de distance en considérant une norme-1, une norme-2 ou une norme infinie. On considère ces 3 normes dans l'étude proposée ici. Elles sont désignées sous les termes `para L1`, `para L2` et `para L`∞.

- pour finir, il a été également proposé par [SOULIER et BOUCARD 2009] d'utiliser une approximation d'une fonction objectif pour chercher la solution d'initialisation dont la réponse associée s'approcherait le plus de la solution approximée au jeu de paramètres considéré. Concernant cette stratégie, on parle plutôt de recherche de la solution dans l'espace des réponses. Elle se formule classiquement de la manière suivante :

$$\mathbf{s}_i^{(0)} = \mathbf{s}(\mathbf{x}^{\text{init}}) \text{ tel que } \mathbf{x}^{\text{init}} = \underset{j \in [\![1, i-1]\!]}{\arg\min} \left| \tilde{y}\left(\mathbf{x}^{(i)}\right) - y\left(\mathbf{x}^{(j)}\right) \right| \tag{2.10}$$

  où $\tilde{y}\left(\mathbf{x}^{(i)}\right)$ est l'approximation de la fonction objectif y au point $\mathbf{x}^{(i)}$ et construite à partir des $i-1$ réponses $y\left(\mathbf{x}^{(j)}\right)$ calculées précédemment pour les jeux de paramètres $\mathbf{x}^{(j)}$ (avec $j \in [\![1, i-1]\!]$). L'approximation peut être obtenue en employant une technique de type métamodèle. On est alors confronté dans ce cas aux problématiques liées à la construction de ce type d'approximation (nombre de points suffisant, bonne répartition des points, qualité d'approximation). Il est également possible de considérer des extensions de cette approche en combinant une recherche dans l'espace des paramètres, dans celui des réponses et dans celui des gradients. La méthode de recherche dans l'espace des réponses sera désignée sous le terme `meta`.

La figure 2.9 illustre les quatre stratégies d'initialisation proposées sur un exemple à une variable.

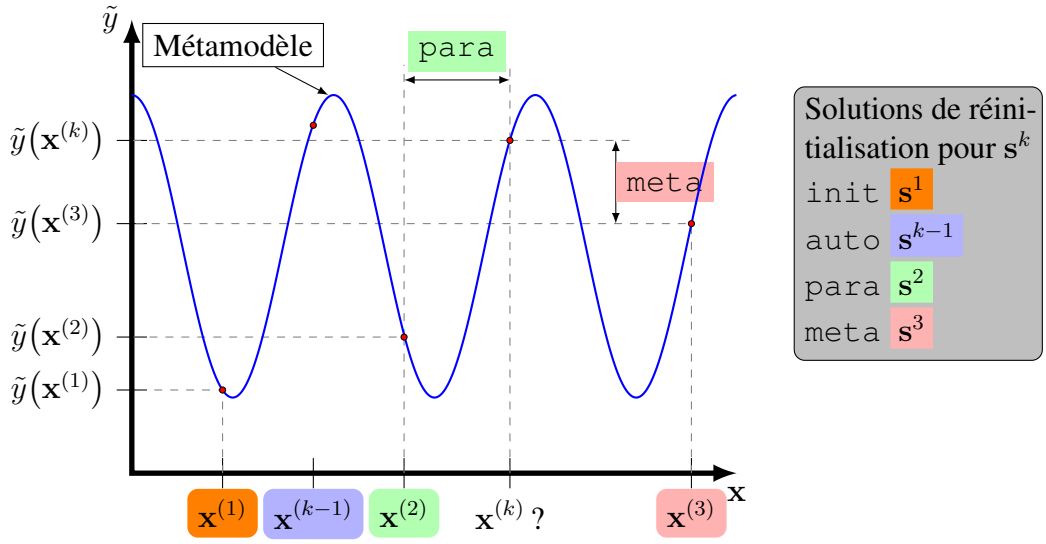


FIGURE 2.9 – Principe des stratégies d'initialisation de l'algorithme LATIN pour obtenir la solution $\mathbf{s}^k$

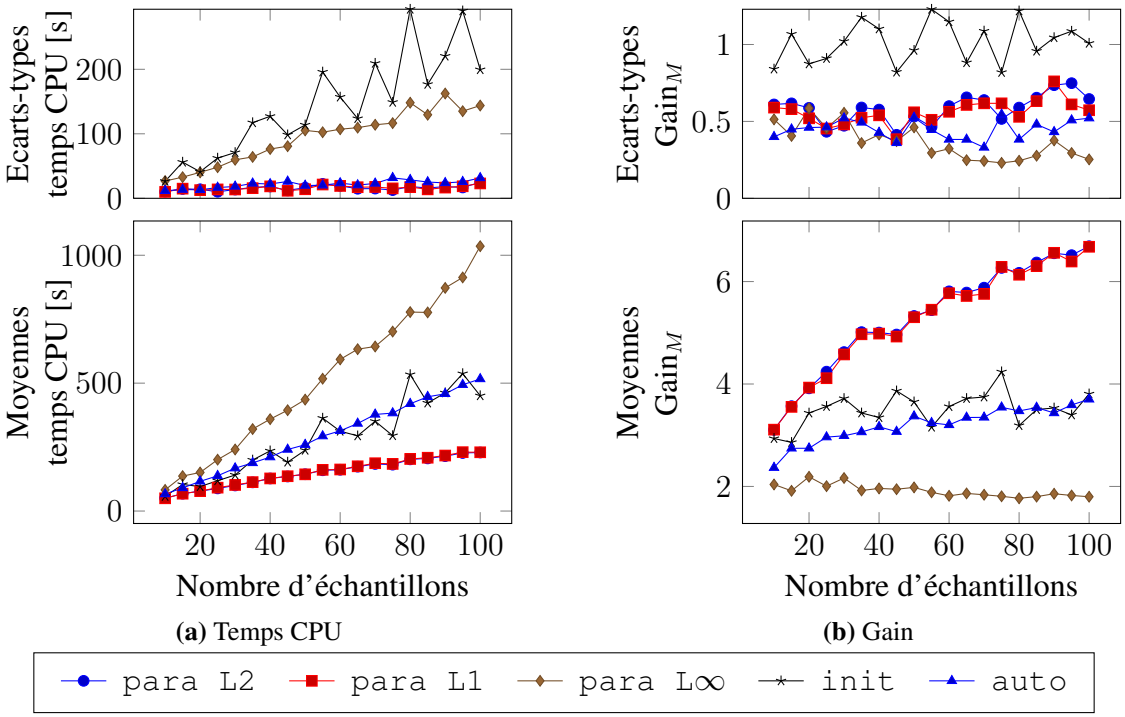
2.3.6 Etude de choix d'une stratégie d'initialisation

Afin de choisir une stratégie d'initialisation de l'algorithme LATIN permettant d'obtenir les meilleures performances, une étude est réalisée. Celle-ci vise à comparer les approches

`init`, `auto` et `para-x` introduites précédemment. Seule l'approche `meta` ne sera pas étudiée mais quelques résultats portant sur cette dernière peuvent être trouvés dans [SOULIER et BOUCARD 2009].

■ **Assemblage considéré :** On considérera pour cette étude le cas-test CSMA présenté en figure 2.2d page 98 dont les spécifications sont données par le tableau 2.1 page 99. Cette fois, deux paramètres seront pris en compte : les deux coefficients de frottement aux interfaces Ω_1-Ω_2 et Ω_2-Ω_3. Ils pourront prendre des valeurs comprises entre 0 et 0.95. Pour l'intégralité de l'étude réalisée ici, le critère d'arrêt de l'algorithme LATIN sera fixé à 10^{-4}.

■ **Résultats :** La figure 2.10 présente les performances en termes de temps de calcul et de gain de la stratégie multiparamétrique pour chaque stratégie de recherche de solution d'initialisation. L'échantillonnage est obtenu en employant une stratégie de tirage de type LHS. On considère des tirages comportant de 10 à 100 jeux de paramètres (par incrément de 5). Pour s'affranchir du caractère aléatoire du tirage LHS, chaque échantillonnage est exécuté 30 fois. La figure présente moyennes et écarts-types des temps CPU et gains pour chaque tirage.



(a) Temps CPU **(b)** Gain

FIGURE 2.10 – Temps CPU et gains obtenus pour 5 stratégies d'initialisation (cas-test `CSMA`, $\eta = 10^{-4}$, 19 valeur de n_s échantillonnés 30 fois chacune par LHS)

■ **Bilan :** L'emploi d'une stratégie d'initialisation de type recherche dans l'espace des paramètres basée sur la norme L2 permet de réduire de manière significative et robuste le temps CPU. Elle permet d'atteindre sur le cas-test considéré un gain s'approchant de 7 pour $n_s = 100$ points échantillonnés. La stratégie d'initialisation basée sur la recherche dans l'espace des paramètres et une norme infinie donne les plus mauvais résultats. L'initialisation de l'algorithme LATIN par la première solution obtenue ne donne pas non plus de résultats aussi significatifs que ceux obtenus par une recherche dans l'espace des paramètres.

Comme l'illustre la figure 2.10b, le gain augmente en fonction du nombre de points échantillonnés. En effet, plus le nombre de points échantillonnés augmente, plus la distance entre les solutions diminue dans l'espace des paramètres. Ce constat est présenté par la figure 2.11a pour deux distances caractéristiques :

- AvgMinDist est la distance minimale intersites moyenne calculée de la manière suivante :

$$\text{AvgMinDist} = \frac{1}{n_s}\sum_{i=1}^{n_s}\min_{j\in[\![1,n_s]\!]}\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\|_2 \tag{2.11}$$

- MinDist est la distance intersites minimale calculée de la manière suivante :

$$\text{MinDist} = \min_{i\in[\![1,n_s]\!]}\min_{j\in[\![1,n_s]\!]}\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\|_2 \tag{2.12}$$

La figure présente la valeur moyenne et l'écart-type de ces deux distances caractéristiques évaluées sur les 30 tirages effectués pour chacune des 19 valeurs de n_s.

La figure 2.11b montre, en utilisant la méthode `para L2`, l'évolution du gain moyen ($\overline{\text{Gain}_M}$) et de son écart-type ($\sigma(\text{Gain}_M)$) calculés pour les 19 valeurs de n_s en fonction de la moyenne sur ces tirages de la distance intersites moyenne. On constate très clairement que le rapprochement global des sites échantillonnés permet d'améliorer la pertinence de l'initialisation de l'algorithme LATIN en employant un critère basé sur une distance dans l'espace des paramètres.

Suite aux constats de cette étude, on emploiera par la suite comme méthode de recherche de la solution d'initialisation la méthode qui consiste en une recherche dans l'espace des paramètres basée sur une norme-2 (désignée sous le terme `para L2` dans cette partie).

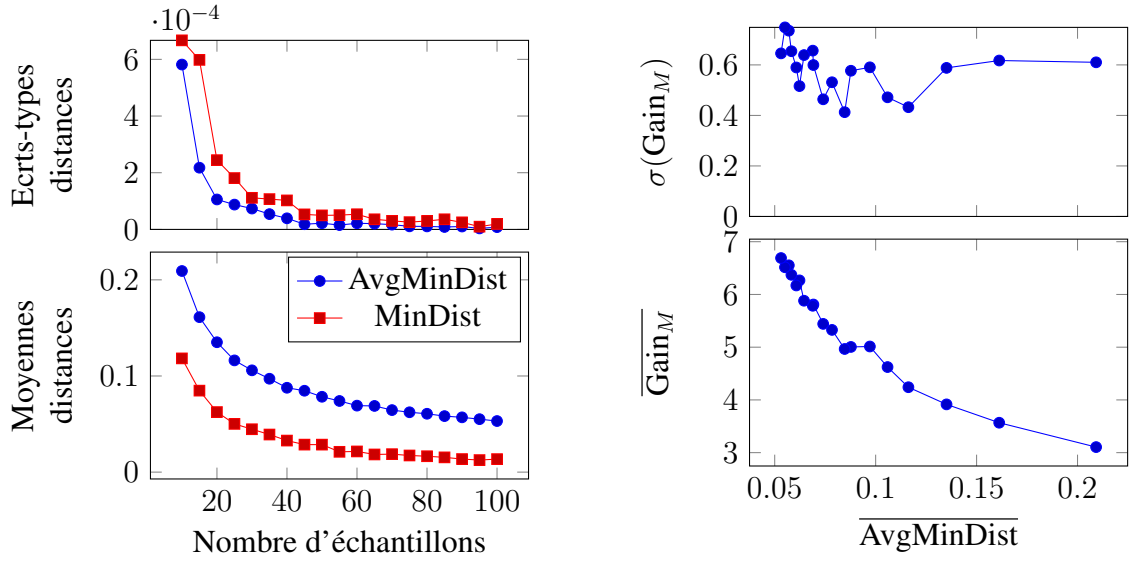
Remarque 2.9 : La notation $\overline{\bullet}$ correspond à la moyenne calculée de la manière suivante :

$$\overline{\bullet} = \overline{\bullet}(n_s) = \frac{1}{n_r}\sum_{i=1}^{n_r}\bullet_i(n_s)$$

où $\bullet_i(n_s)$ correspond i-ème valeur de la grandeur $\bullet$ obtenue à n_s fixé pour un nouveau tirage obtenu par LHS (le tirage sera exécuté n_r fois à n_s fixé).

La notation $\sigma(\bullet)$ correspond à l'écart-type calculé de la manière suivante :

$$\sigma(\bullet) = \sigma(\bullet, n_s) = \sqrt{\frac{1}{n_r}\sum_{i=1}^{n_r}\left(\bullet_i(n_s)-\overline{\bullet}(n_s)\right)^2}$$



(a) Evolution des distances minimales intersites en fonction du nombre d'échantillons

(b) Evolution du gain en fonction de la moyenne des distances minimales intersites

FIGURE 2.11 – Evolutions des distances intersites et dépendance du gain de la stratégie multiparamétrique à la moyenne des distance intersites (cas-test `CSMA`, $\eta = 10^{-4}$, 19 valeur de n_s échantillonnés 30 fois chacune par LHS, initialisation `para L2`)

2.3.7 Méthodes de tri des jeux de paramètres

Dans la mesure où la proximité de la solution d'initialisation dans l'espace des paramètres permet de réduire les temps de calcul, on s'intéresse dans cette partie à la mise en œuvre d'une méthode de tri des points échantillonnés. L'objectif consiste ici à trouver une méthode de parcours des échantillons permettant de minimiser le coût de calcul et de maximiser le gain. Pour faciliter la compréhension, on notera $\mathbf{x}$ les jeux de paramètres non encore triés et $\hat{\mathbf{x}}$ les paramètres qui le sont. Ainsi, plusieurs méthodes de tri des échantillons sont considérées :

- la première approche consiste à trier les échantillons en minimisant les variations selon un paramètre. En effet, dans certains cas, l'impact sur les performances des variations de certains paramètres entre deux résolutions peut être très important. Ce constat avait déjà été fait dans la partie 2.3.2 page 100 où, selon le type de paramètres, les performances de la stratégie multiparamétrique en termes de temps de calcul et de gain pouvaient être très variables. Cette approche ne sera pas étudiée ici car les paramètres pris en compte sont du même type. Pour un assemblage présentant par exemple des paramètres de type jeu et frottement, on cherchera à minimiser les variations du paramètre de type jeu.

- afin de minimiser les variations entre les paramètres de deux résolutions consécutives, il est possible de trier les tirages à partir d'un jeu de paramètres initial $\mathbf{x}_{ref}$ puis de chercher le point le plus proche dans l'espace des paramètres en employant une

norme-2 (par exemple). Le tri est alors réalisé au fur et à mesure de la manière suivante :

$$\hat{\mathbf{x}}^{(1)} = \min_{j \in [\![1, n_s]\!]} \left\| \mathbf{x}_{ref} - \mathbf{x}^{(j)} \right\|_2 \tag{2.13}$$

$$\forall i \in [\![2, n_s]\!],\ \hat{\mathbf{x}}^{(i)} = \min_{j \in [\![i, n_s]\!]} \left\| \hat{\mathbf{x}}^{(i-1)} - \mathbf{x}^{(j)} \right\|_2 \tag{2.14}$$

Cette méthode sera désignée sous le terme `proche-en-proche` (`pep`).

- le principal défaut de cette méthode réside dans le fait qu'elle peut engendrer un ensemble de jeux de paramètres présentant une variation importante des paramètres dans le cas où l'échantillonnage initial ne serait pas suffisamment uniformément réparti dans l'espace des paramètres. Pour remédier à ce problème, une stratégie de tri basée sur l'ensemble des points déjà triés est proposée. Le jeu de paramètres i n'est pas cherché au plus près du point $i-1$ mais au plus près du barycentre des $i-1$ points déjà triés tout en gravitant autour d'un jeu de paramètres de référence $\mathbf{x}_{ref}$. Ainsi, le jeu de paramètres $\hat{\mathbf{x}}^{(i)}$ est obtenu de la manière suivante :

$$\hat{\mathbf{x}}^{(1)} = \min_{j \in [\![1, n_s]\!]} \left\| \mathbf{x}_{ref} - \mathbf{x}^{(j)} \right\|_2 \tag{2.15}$$

$$\forall i \in [\![2, n_s]\!],\ \hat{\mathbf{x}}^{(i)} = \min_{j \in [\![i, n_s]\!]} \left\| \mathbf{b}^{(i-1)} - \mathbf{x}^{(j)} \right\|_2$$

$$\text{où } \mathbf{b}^{(i-1)} = \mathbf{x}_{ref} + \frac{1}{2n_{i-1}} \sum_{k=1}^{n_{i-1}} \hat{\mathbf{x}}^{(k)} \tag{2.16}$$

où n_{i-1} est le nombre de paramètres qui ont déjà été triés. Cette méthode sera désignée sous le terme `proche-bary` (`pb`).

■ **Application sur un exemple :** En considérant le cas-test CSMA à deux paramètres, on emploie un ensemble de méthodes de tri issues de celles présentées précédemment en choisissant différents points de référence $\mathbf{x}_{ref}$. Les différentes combinaisons étudiées sont présentées dans le tableau 2.3.

Les méthodes `xx-sc` et `xx-c` (`xx`= `pb` ou `pep`) consistent à prendre comme point de référence $\mathbf{x}_{ref}$ respectivement le centre de l'espace de conception $\mathcal{D}$ où varient les paramètres et le barycentre des points échantillonnés $\mathbf{x}^{(i)}$. Les deux méthodes `pb-`i et `pep-`i consistent quant à elles à prendre les points $\mathbf{x}_r^{(i)}$ choisis manuellement dans l'espace de conception et dont les valeurs sont données par le tableau 2.4. Les méthodes `pb-init` et `pep-init` sont obtenues en considérant comme point de référence le premier jeu de paramètres obtenu par l'échantillonnage avant le tri.

Ces 4 jeux de paramètres correspondent aux 4 coins de l'espace de conception et permettent d'obtenir la première solution par l'algorithme LATIN associé à 4 types de comportement de l'assemblage. Ainsi, tel que le montre la figure 2.12 qui présente l'allure de l'effort à la paroi en fonction des deux coefficients de frottement, les 4 jeux de paramètres

	proche-en-proche	proche-bary	$\mathbf{x}_{ref}$
pb-init		✓	$\mathbf{x}^{(1)}$
pep-init	✓		
pb-sc		✓	centre $\mathcal{D}$
pep-sc	✓		
pb-c		✓	barycentre échantillons
pep-c	✓		
pb-i		✓	$\mathbf{x}_r^{(i)}$
pep-i	✓		

TABLE 2.3 – Stratégies de tri des tirages

	Jeux de paramètres	
$\mathbf{x}_r^{(1)}$	$\mu_1 = 0$	$\mu_2 = 0$
$\mathbf{x}_r^{(2)}$	$\mu_1 = 0$	$\mu_2 = 0.95$
$\mathbf{x}_r^{(3)}$	$\mu_1 = 0.95$	$\mu_2 = 0.95$
$\mathbf{x}_r^{(4)}$	$\mu_1 = 0.95$	$\mu_2 = 0$

TABLE 2.4 – 4 points de référence considérés

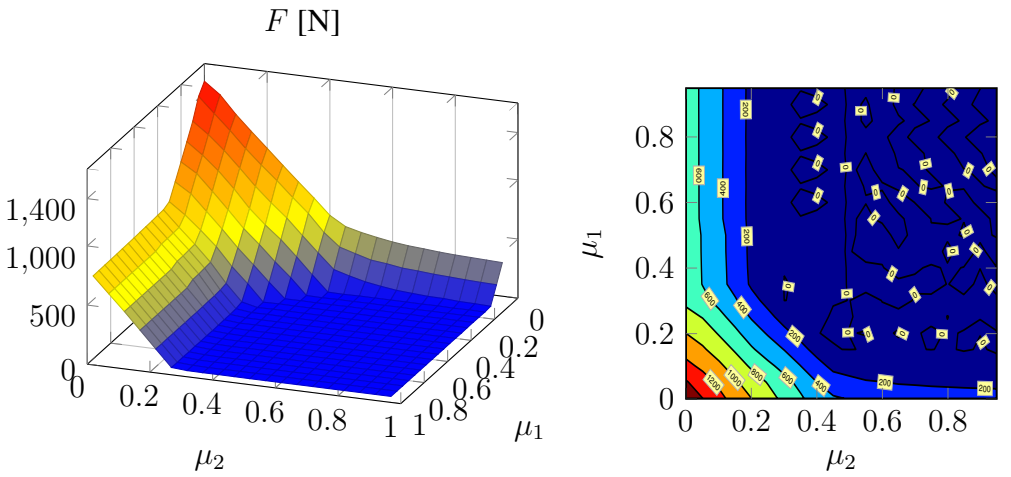
correspondent pour $\mathbf{x}_r^{(3)}$ à une phase sans contact à la paroi (effort nul) et pour les 3 autres jeux de paramètres des phases de contact à la paroi avec ou sans glissement aux interfaces $\Omega_1 - \Omega_2$ et $\Omega_2 - \Omega_3$.

Afin d'étudier l'impact de ces méthodes de tri des jeux de paramètres, les 30×19 tirages LHS utilisés précédemment sont triés, puis la stratégie multiparamétrique est employée pour obtenir les solutions associées. Les performances en termes de temps de calcul et de gains sont regroupées dans la figure 2.13.

Comme on l'observe sur les figures 2.13a et 2.13b, les méthodes de tri n'ont quasiment pas d'impact sur les performances aussi bien en termes de temps de calcul que de gain. On constate cependant en observant les écarts-types que les approches pb-sc et pb-c permettent d'obtenir une robustesse légèrement plus importante que les autres approches du point de vue des performances.

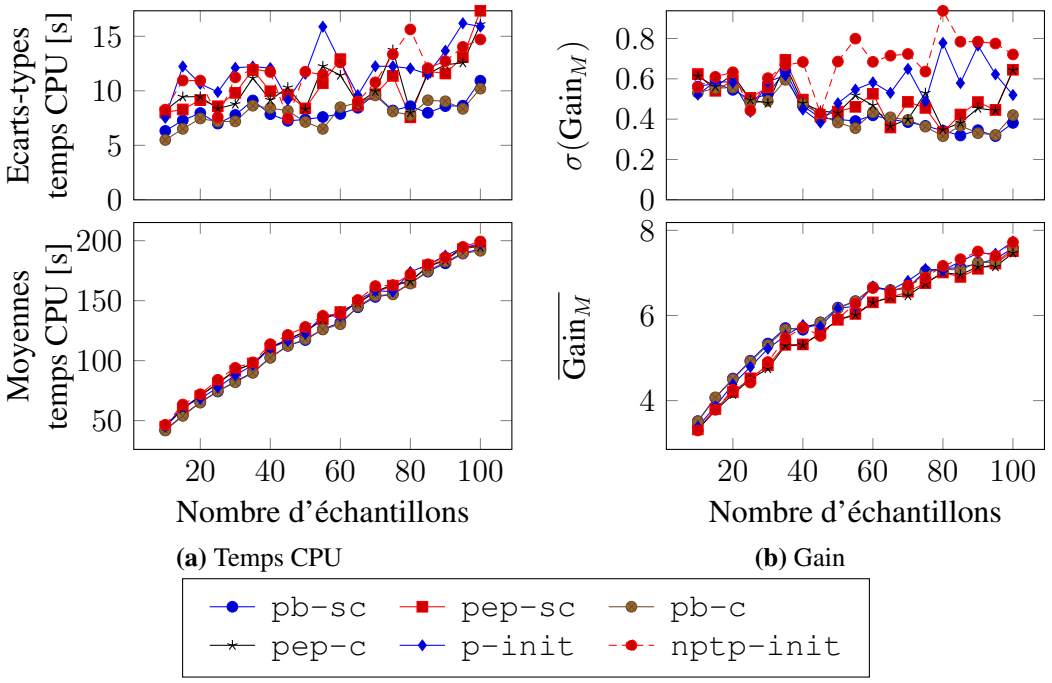
Remarque 2.10 : Les méthodes de tri présentées ici sont bien entendu appliquées sur des jeux de paramètres qui ont été normés au préalable pour tenir compte les problèmes liés aux ordres de grandeur différents.

■ **Choix du premier jeu de paramètres :** On s'intéresse ici à l'impact du choix du premier calcul sur les performances de la résolution. On considère pour cela les approches pb-i et pep-i présentées dans le tableau 2.3 construites à partir des jeux de paramètres de références issus du tableau 2.4. Les performances en termes de temps de calcul et de gains sont visibles dans la figure 2.14.



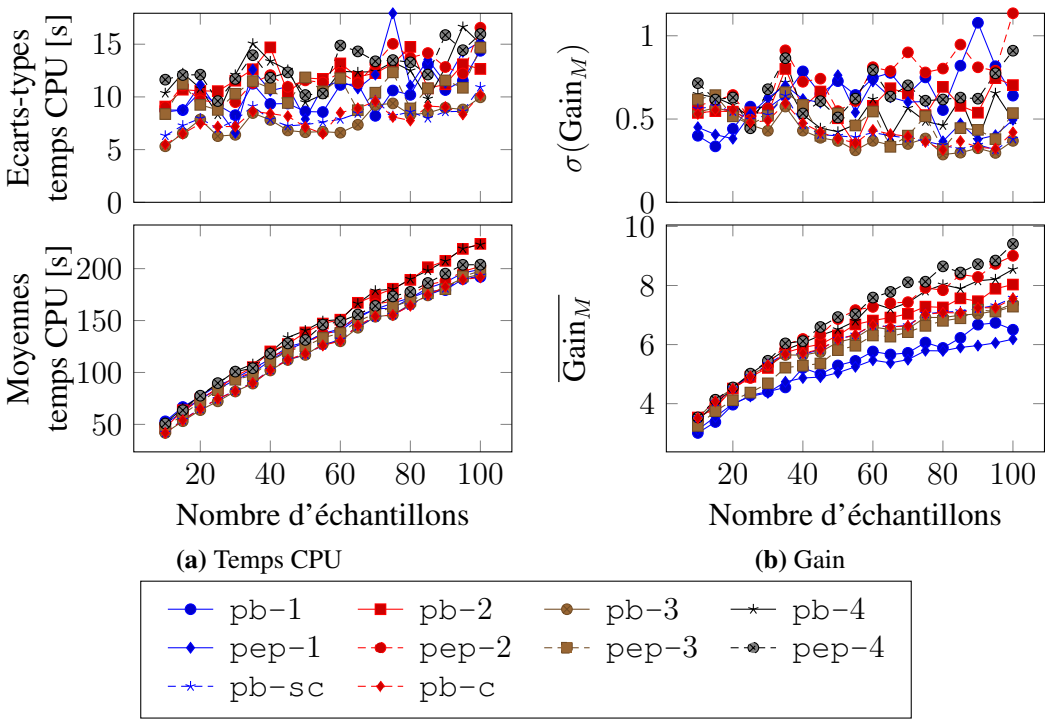
(a) **(b)**

FIGURE 2.12 – Valeur de l'effort à la paroi en fonction des paramètres pour le cas-test `CSMA` à 2 paramètres (figure obtenue par une grille régulière 20×20 dans l'espace des paramètres)



(a) Temps CPU **(b)** Gain

FIGURE 2.13 – Temps CPU et gain de la stratégie multiparamétrique pour différentes méthodes de tri des paramètres (cas-test `CSMA`, 2 paramètres, $\eta = 10^{-4}$, 19 valeurs de n_s échantillonnées 30 fois chacune par LHS, initialisation `para L2`)



(a) Temps CPU **(b)** Gain

FIGURE 2.14 – Temps CPU et gain de la stratégie multiparamétrique pour différentes méthodes de tri des paramètres et pour différents jeux de paramètres de référence $\mathbf{x}_{ref}$ (cas-test `CSMA`, 2 paramètres, $\eta = 10^{-4}$, 19 valeurs de n_s échantillonnées 30 fois chacune par LHS, initialisation `para L2`)

Cette fois, on constate que l'impact du choix du jeu de paramètres initial sur le temps de calcul est faible mais non négligeable. Du point de vue du gain, l'effet est plus marqué. En effet, diminuer globalement le temps de calcul signifie que globalement chaque résolution est obtenue avec un temps de calcul plus faible. Ainsi le temps CPU de référence considéré pour le calcul du gain est moins élevé et la valeur du gain s'en trouve diminuée (l'écart entre le temps CPU de référence et celui des autres résolutions est plus faible).

Finalement, il s'avère extrêmement complexe de trouver un jeu de paramètres de référence permettant de réduire de manière significative les temps de calcul. Ce constat est d'autant plus vrai que, dans la pratique, le comportement de l'assemblage en fonction des valeurs de paramètres est souvent impossible à prédire.

2.4 Bilan intermédiaire :

A l'issue de ces études de performances, l'élément clé réside dans le choix de la stratégie d'initialisation de la stratégie multiparamétrique. L'étude proposée ici montre que la stratégie multiparamétrique présente une efficacité maximale lorsqu'elle se base sur une initialisation de l'algorithme LATIN à partir d'une solution précédemment convergée proche, au sens d'une métrique dans l'espace des paramètres, du jeu de paramètres pour lequel on cherche une nouvelle solution. C'est pourquoi, dans la suite, seule la stratégie multiparamétrique basée sur la méthode `para L2` sera employée. Par ailleurs, procéder à un tri des jeux de paramètres n'apporte pas de réelle amélioration des performances, si ce n'est une légère amélioration de leur robustesse en employant une méthode de tri visant à progresser autour d'un jeu de paramètres de référence (méthode de type `pb-c` ou `pb-sc`). Par la suite, lorsque l'ensemble des jeux de paramètres est connu *a priori*, ce type de méthode sera systématiquement utilisé.

3 Calcul des gradients

Dans le cadre de l'optimisation et plus généralement des études paramétriques, de nombreux algorithmes ont besoin des gradients de la fonction objectif. D'un point de vue optimisation par exemple, de nombreux algorithmes nécessitent cette information. Au vu des performances de la stratégie multiparamétrique présentées dans la partie précédente, il est proposé de tirer profit de celles-ci afin de calculer à moindre coût les gradients.

Dans cette partie, les méthodes classiques permettant de calculer les gradients seront tout d'abord brièvement introduites. Ensuite, une démarche de calcul de gradients adaptée à la stratégie multiparamétrique sera proposée et ses performances étudiées.

3.1 Méthodes de calcul des gradients

Le nombre de méthodes permettant l'obtention des gradients d'une fonction calculée à partir de la résolution d'un problème de mécanique est important. Elles nécessitent une mise en œuvre plus ou moins intrusive et complexe [HAFTKA et ADELMAN 1989 ; TOLSMA

et P. I. BARTON 1998 ; van KEULEN, HAFTKA et QU 2000 ; van KEULEN, HAFTKA et N. KIM 2005]. On propose dans cette partie de rappeler brièvement les méthodes de calcul des gradients.

3.1.1 Différences finies

La méthode des différences finies [BOOLE et MOULTON 1880] est à la fois la plus ancienne méthode permettant d'approcher les dérivées d'une fonction mais aussi *a priori* la plus simple à utiliser. D'apparence extrêmement simple de principe, le calcul des différences finies s'avère dans la pratique relativement compliqué, en particulier lorsque l'on cherche à l'appliquer au calcul des dérivées d'une fonction bruitée. En effet, elle recèle un paramètre que l'on nommera **pas** dont le choix peut s'avérer complexe.

■ **Principe :** L'idée maîtresse des différences finies repose sur l'idée que, par définition, la dérivée d'une fonction d'une variable s'écrit :

Définition 2.1 (Dérivée) **:** Soit y une fonction de E dans $\mathbb{R}$, on dit que f est dérivable au point $x \in E$ si et seulement si le taux d'accroissement $\dfrac{y(x+h)-y(x)}{h}$ admet une limite finie lorsque h tend vers 0. On appelle dérivée de y en 0, la grandeur $y'(x)$ définie telle que

$$y'(x) = \lim_{h \to 0} \frac{y(x+h)-y(x)}{h}$$

Lorsque la fonction y est connue de façon explicite, les valeurs des dérivées de la fonction peuvent être déterminées de manière plus ou moins aisée à partir des dérivées connues de fonctions usuelles. Dans le cas qui est le nôtre, la fonction n'étant pas connue de manière explicite, il n'est pas possible de calculer directement les dérivées. Pour remédier à ce problème, une solution consiste à choisir une valeur de h puis à calculer $y(x+h)$ et $y(x)$. Afin de retrouver la définition ci-dessous, il « suffit » alors de faire tendre h vers 0.

Dans ce contexte, l'emploi des développements en série de Taylor conduit à

$$y(x+h) \underset{h \to 0}{=} y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \ldots = \sum_{n=0}^{+\infty} \frac{h^n}{n!} y^{(n)}(x) \tag{2.17}$$

$$= \sum_{n=0}^{N} \frac{h^n}{n!} y^{(n)}(x) + O(h^N) \tag{2.18}$$

La restriction du développement à un ordre fini N permet l'obtention de la différence $y(x+h)-y(x)$ et ainsi d'une approximation à l'ordre N de la dérivée de la fonction y. On parlera dans ce cas de différence finie avancée.

Il est également possible d'utiliser les développements de Taylor de $y(x-h)$ ou de $y(x+2h')$ et $y(x-2h')$ lorsque h et h' tendent vers 0. Les approximations de la dérivée

seront alors respectivement désignées sous le terme de différence finie rétrograde et différence finie centrée [5].

Ainsi au premier ordre, on obtient les différences finies suivantes :

Différence finie avancée (FD1)	$y'(x) = \dfrac{y(x+h) - y(x)}{h} + O(h)$
Différence finie rétrograde (BD1)	$y'(x) = \dfrac{y(x) - y(x-h)}{h} + O(h)$

Au second ordre, il est également possible d'obtenir la dérivée de la fonction y par différence centrée (CD2) :

$$y'(x) = \frac{y(x+h) - y(x-h)}{2h} + O(h^2)$$

Par ailleurs, les formules des différences finies sont généralisables à la dérivée n-ième de la fonction y.

En posant $x_i = x + ih$ et $y_i = y(x_i)$, on peut construire le tableau 2.5 permettant de construire les approximations par différences finies des dérivées premières et secondes.

		y_{-3}	y_{-2}	y_{-1}	y_0	y_1	y_2	y_3	n_s
FD1	$hy'(x)$				-1	1			$n_p + 1$
	$h^2y''(x)$				1	-2	1		$2n_p + 1$
BD1	$hy'(x)$			-1	1				$n_p + 1$
	$h^2y''(x)$		1	-2	1				$2n_p + 1$
FD2	$2hy'(x)$				-3	4	-1		$2n_p + 1$
	$h^2y''(x)$	-1	4	-5	2				$3n_p + 1$
BD2	$2hy'(x)$		1	-4	3				$2n_p + 1$
	$h^2y''(x)$				2	-5	4	-1	$3n_p + 1$
CD2	$2hy'(x)$			-1		1			$2n_p$
	$h^2y''(x)$			1	-2	1			$2n_p + 1$
CD4	$12hy'(x)$		1	-8		8	-1		$4n_p$
	$12h^2y''(x)$		-1	16	-30	16	-1		$4n_p + 1$

TABLE 2.5 – Coefficients de calcul des différences finies

Comme le montre la dernière colonne du tableau, l'augmentation de l'ordre des différences finies engendre une augmentation du nombre d'évaluations de la fonction pour approximer les gradients. Par ailleurs, ce nombre croît également avec le nombre de paramètres n_p du problème.

5. Pour des raisons pratiques, les notations FDx, BDx et CDx seront utilisées par la suite pour désigner respectivement les différences finies avancées, rétrogrades et centrées à l'ordre x (*Forward Difference, Backward Difference, Centered Difference*).

Remarque 2.11 : Généralement, dans le cadre de l'optimisation, la réponse de la fonction est évaluée en x. Le calcul des gradients employant des schémas centrés ne tire pas partie du fait que cette réponse est disponible.

Remarque 2.12 : La présence d'un point x en bordure de l'espace des paramètres peut rendre le calcul de la dérivée impossible. Dans ce cas, on corrigera le type de dérivée prise en compte par un autre type permettant le calcul (si le calcul par CD2 est impossible, on se rabattra sur les formulations FD1 ou BD1).

■ **Erreurs présentes lors du calcul des gradients :** A l'instar des travaux de [GILL et al. 1981 ; GILL et al. 1983], posons $\varphi(y,h)$ l'approximation par différence finie de la dérivée y'. Dans le contexte des différences finies, 3 sources d'erreurs peuvent être identifiées :

- une **erreur de troncature** $T(h)$ liée au fait que l'on néglige des termes du développement en séries de Taylor. Dans le cadre des différences finies avancées à l'ordre 1 [6], elle est définie telle que :

$$\varphi_F(y,h) - y'(x) = \sum_{\mathbf{n=1}}^{+\infty} \frac{h^n}{n!} y^{(n)}(x) = \frac{h}{2} y''(\xi) = T_F(h) \text{ où } \xi \in [x, x+h] \quad (2.19)$$

- dans la pratique, la précision finie des machines ne permet pas de calculer exactement la valeur de la fonction y en x et $x+h$. Ainsi, il est possible de réécrire la valeur de la fonction y sous la forme $\hat{y}$ en y incluant une erreur $\epsilon(x)$ telle que $\hat{y}(x) = y(x) + \epsilon(x)$. Un majorant de cette erreur pourra être noté ϵ_A tel que $|\epsilon(x)| \leq \epsilon_A$. Ainsi, en posant $|\theta_0| \leq 1$ et $|\theta_h| \leq 1$, on peut formuler la réponse en x et $x+h$ de la fonction y obtenue par un solveur numérique sous la forme :

$$\hat{y}(x) = y(x) + \theta_0 \epsilon_A \text{ et } \hat{y}(x+h) = y(x+h) + \theta_h \epsilon_A \quad (2.20)$$

  En tenant compte de ces erreurs, il est possible de construire l'**erreur de calcul** [DUMONTET et VIGNES 1977] (*condition error* [LYNESS 1977] ou *cancellation error*) $C(h)$ [7] définie en considérant $|\psi| \leq 1$ telle que

$$\varphi_F(\hat{y},h) - \varphi_F(y,h) = \frac{2\psi\epsilon_A}{h} = C_F(h) \text{ d'où } |C_F(h)| \leq \frac{2\epsilon_A}{h} \quad (2.21)$$

- une **erreur d'arrondi** intervenant dans le calcul de $\varphi(y,h)$. Cette erreur intervient dans le calcul des soustractions et divisions lors de l'évaluation des différences finies. Généralement, cette erreur, qui est due à la précision machine, est négligeable par rapport au deux autres erreurs et n'est pas prise en compte.

6. Dans ce cas, l'approximation par différence finie et l'erreur de troncature associée seront respectivement notées $\varphi_F(y,h)$ et $T_F(h)$

7. Notée $C_F(h)$ dans le cadre des différences finies avancées à l'ordre 1 tel que présenté ici.

Finalement, l'erreur de calcul de l'approximation de la dérivée par différence finie se résume à la somme de l'erreur de troncature et de celle d'annulation. Un majorant de la valeur absolue de cette erreur sera noté E [8] et définie telle que

$$E_F = \frac{h}{2}\left|y''(\xi)\right| + \frac{2}{h}\epsilon_A \tag{2.22}$$

La principale difficulté réside dans l'obtention de la grandeur ϵ_A. Des procédures pour l'obtenir peuvent être trouvées dans [HAMMING 1973 ; GILL, MURRAY et M. WRIGHT 1981] et [IOTT, HAFTKA et ADELMAN 1985] en annexe. Plus récemment, des travaux proposent sa détermination à partir de l'évaluation du bruit numérique [MORÉ et WILD 2011].

Remarque 2.13 : Les éléments succincts présentés ci-dessus peuvent bien entendu être obtenus pour des différences finies rétrogrades et centrées et à des ordres plus élevés (voir par exemple [GILL et al. 1983]).

■ **Choix du pas et qualité de l'approximation du gradient :** La principale problématique liée à la mise en œuvre des différences finies réside dans le choix du pas. En effet, l'idée de base consiste à choisir un pas h le plus faible possible (l'idéal étant de faire tendre h vers 0). Or, dans la pratique, la précision numérique de l'évaluation de la réponse ne permet pas de choisir un pas trop faible (l'erreur d'annulation augmente). Dans le cas des calculs d'assemblages et plus généralement des calculs mécaniques, de nombreux facteurs peuvent impacter cette erreur. On pourra citer par exemple le maillage, le critère d'arrêt de l'algorithme LATIN...

Par ailleurs, il est également important d'être vigilant quant à la valeur que prend le pas lorsqu'on l'augmente. En effet, dans ce cas, l'erreur de troncature augmente. On se trouve alors dans une situation souvent désignée sous le terme de *step-size dilemma* (dilemme du pas). Pour remédier à ce problème, une solution consiste à minimiser l'erreur E_F introduite par l'équation 2.22. Cette minimisation conduit alors au pas « optimal » :

$$h_F^* = 2\sqrt{\frac{\epsilon_A}{\left|y''(\xi)\right|}} \tag{2.23}$$

L'obtention du terme $y''(\xi)$ n'étant pas faisable directement, [GILL et al. 1983] proposent de l'approximer par un schéma de différence finie centrée à l'ordre 2 et élaborent un algorithme (*algorithm FD*) permettant l'estimation automatique du pas, des dérivées première et seconde tout en minimisant l'erreur d'approximation de la dérivée. On trouve des exemples de mise en œuvre de cette méthode dans [R. R. BARTON 1990] et une approche de détermination du pas similaire dans [BRÉZILLON et al. 1981].

Des éléments de discussion concernant le calcul des dérivées d'une fonction bruitée sont disponibles dans [MORÉ et WILD 2012a ; MORÉ et WILD 2012b].

La difficultés de détermination du pas ainsi que la nécessité d'un nombre important d'évaluations de la fonction objectif pour obtenir un gradient de qualité ont conduit, au

8. Notée dans le cadre des différences finies à l'ordre 1 E_F.

début des années 1990, à l'élaboration de nouvelles méthodes permettant de calculer les gradients.

3.1.2 Approche par variable complexe

Proposée par [SQUIRE et TRAPP 1998] d'après les travaux de [LYNESS et MOLER 1967], cette approche consiste à estimer les dérivées de la fonction y à variable et à valeur dans $\mathbb{R}$ en la considérant comme une fonction à variable complexe. Ainsi, le développement en série de Taylor de y s'écrit en considérant le nombre complexe i :

$$y(x+ih) = y(x) + ihy'(x) - \frac{h^2}{2}y''(x) - i\frac{h^3}{3!}y^{(3)}(x) + \dots \tag{2.24}$$

Le calcul de la partie réelle et de la partie imaginaire de $y(x+ih)$ conduit à

$$\text{Re}\left[y(x+ih)\right] = y(x) - \frac{h^2}{2}y''(x) + O(h^4) \tag{2.25}$$

$$\text{Im}\left[y(x+ih)\right] = hy'(x) - \frac{h^3}{3!}y^{(3)}(x) + O(h^5) \tag{2.26}$$

On obtient finalement, en divisant cette dernière équation par h, à l'ordre 2 :

$$y'(x) = \frac{\text{Im}\left[y(x+ih)\right]}{h} + O(h^2) \tag{2.27}$$

L'avantage de la formulation présentée par l'équation 2.27 est qu'elle permet l'obtention d'une approximation de la dérivée première sans opération de soustraction. L'erreur de calcul n'est donc plus présente et l'erreur de l'approximation se limite alors à l'erreur liée à la troncature de la série de Taylor. Le pas peut donc être choisi petit sans entraîner de perte de précision de l'estimation de la dérivée. Cet avantage n'est plus vrai dans le cas de la dérivée seconde qui, à partir des équations 2.25 et 2.26, peut être estimée à l'ordre 2 de la manière suivante :

$$y''(x) = \frac{2\left(y(x) - \text{Re}\left[y(x+ih)\right]\right)}{h^2} + O(h^2) \tag{2.28}$$

Dans ce cas, la présence de la soustraction engendre à nouveau une erreur de calcul. Les travaux [FIKE et ALONSO 2011] proposent une nouvelle formulation qui permet de résoudre ce problème.

Le principal défaut de cette approche est qu'elle nécessite l'accès au code source de la fonction dont on cherche les gradients. En effet, un certain nombre de modifications doivent y être faites :

- les variables réelles du code doivent être remplacées par des variables complexes ;
- toutes les fonctions et les opérateurs doivent être redéfinis pour tenir compte des variables complexes.

Ces modifications peuvent s'avérer lourde. En particulier certain opérateurs ne sont pas définis pour les complexes (les inégalités par exemple). [MARTINS 2002] s'intéresse en détail à ces problématiques en proposant des solutions adaptées au passage aux variables complexes et à la redéfinition des opérateurs et fonctions. Des exemples d'application de cette approche sur des exemples de calculs fluide, solide et aéroélastique peuvent être trouvés dans [MARTINS, STURDZA et ALONSO 2003 ; MARTINS, ALONSO et REUTHER 2004] et des comparaisons avec d'autres méthodes sont développées dans [KEANE et NAIR 2005].

Remarque 2.14 : Il ne peut être ignoré que cette approche nécessite le double de mémoire par rapport à une approche classique (basée sur des variables réelles). En effet, il est nécessaire de stocker, pour toutes les grandeurs, les parties réelles et imaginaires. Par ailleurs, dans les cas présentés par les auteurs précédents, le temps de résolution du problème avec variables complexes est environ 3 fois plus important que celui du problème initial.

3.1.3 Méthode directe

Beaucoup plus intrusives que les méthodes précédentes, les méthodes directes proposent d'intégrer les dérivées au cœur même de la formulation modélisée du problème physique considéré. Ainsi, en considérant un problème discret, comme c'est le cas dans les travaux de cette thèse, réduit à la formulation [KEANE et NAIR 2005] :

$$\mathbf{R}\left(\mathbf{x}, \mathbf{w}(\mathbf{x})\right) = \mathbf{0} \tag{2.29}$$

Classiquement obtenu à partir d'équations aux dérivées partielles, $\mathbf{R}$ peut être vu comme le résidu issu de la formulation matricielle du problème discrétisé. Ce dernier peut être dépendant de manière explicite aux n_p paramètres de conception $\mathbf{x}$ ou implicite représenté ici par l'intermédiaire de la fonction $\mathbf{w}(\mathbf{x})$.

L'objectif ici serait de déterminer les gradients d'une fonction $y = y(\mathbf{x}, \mathbf{w}(\mathbf{x}))$ soit de calculer :

$$\frac{\mathrm{d}y}{\mathrm{d}x_i} = \frac{\partial y}{\partial x_i} + \frac{\partial y}{\partial \mathbf{w}} \frac{\mathrm{d}\mathbf{w}}{\mathrm{d}x_i} \tag{2.30}$$

La dérivation par rapport aux paramètres de conception de l'équation 2.29 conduit à

$$\forall i \in [\![1, n_p]\!],\ \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \frac{\mathrm{d}\mathbf{w}}{\mathrm{d}x_i} + \frac{\partial \mathbf{R}}{\partial x_i} = \mathbf{0} \tag{2.31}$$

La résolution du système consiste à résoudre n_p fois l'équation 2.31 (pour chaque paramètre) afin d'obtenir le gradient de y. La taille du problème complet devient alors extrêmement importante.

Pour des compléments sur les méthodes directes, on pourra se référer à [ADELMAN et HAFTKA 1986 ; BARTHELEMY, HAFTKA et G. A. COHEN 1989 ; HAFTKA et ADELMAN 1989] et trouver quelques exemples simples (problèmes de statique, de valeurs propres et de dynamique transitoire) de mise en œuvre dans [KEANE et NAIR 2005].

3.1.4 Méthode adjointe

L'approche directe présentée précédemment s'avère non viable dans le cas où le problème physique discrétisé comporte un nombre important de degrés de liberté ou lorsque le problème comporte un nombre important de paramètres. En effet, le coût de calcul nécessaire à la résolution du problème et à l'obtention des gradients devient prohibitif. C'est pourquoi, il a été proposé dans les années 1980 d'employer une méthode dite **adjointe** pour obtenir les gradients [CACUCI 1981]. Les méthodes adjointes ont été utilisées initialement pour traiter des problèmes inverses et de contrôle optimal [LIONS 1971]. Elles furent par la suite employées dans le contexte de la conception [PIRONNEAU 1974] et plus récemment dans le cadre de la conception aéronautique (par exemple [JAMESON 1988 ; JAMESON 2001 ; JAMESON et S. KIM 2003]).

Classiquement, pour le calcul des gradients, on identifie deux stratégies : la première est une formulation discrète du problème adjoint qui, comme son nom l'indique, se base sur la formulation discrétisée du système physique modélisé et s'articule autour de l'écriture d'un Lagrangien. La seconde approche, quant à elle, se base sur la formulation continue du problème. Cette dernière présente de nombreuses difficultés inhérentes à la formulation du problème aux dérivées correspondant au problème initial. On pourra se référer à [GILES et PIERCE 2000] pour des détails sur la formulation continue et [KEANE et NAIR 2005] pour des compléments et détails généraux sur les méthodes adjointes pour le calcul des dérivées.

3.1.5 Différenciation automatique

La méthode de différenciation automatique (*Automatic Differentiation*) diffère grandement des précédentes méthodes. En effet, elle ne propose plus d'agir au niveau de la modélisation continue ou discrète du problème ni d'agir de manière non-intrusive, mais se concentre sur le code source du programme employé pour résoudre le problème. Sa mise en œuvre nécessite le code source de l'intégralité du programme (y compris les librairies).

Le principe de la différenciation automatique [BEDA et al. 1959 ; WENGERT 1964] part du constat de la structuration d'un programme informatique. Ce dernier est constitué d'un ensemble d'opérations arithmétiques simples telles que des additions, des soustractions, des multiplications et des appels à des fonctions intrinsèquement définies (comme les fonctions `sin`, `cos`, `exp` ...). La différenciation automatique remonte jusqu'à ces opérations élémentaires pour concevoir, en employant des règles de calcul différentiel, un nouveau programme capable de calculer les dérivées par rapport aux paramètres d'une fonction d'intérêt fournie par le programme initial. Des éléments relatifs à la différentiation automatique peuvent être trouvés dans [GRIEWANK 1989].

La technicité nécessaire à la mise en œuvre de la différentiation automatique est directement liée à la complexité du programme original et aux types d'opérations qui le composent. En particulier, certains opérateurs peuvent s'avérer particulièrement complexes à prendre en compte (les inégalités par exemple). Par ailleurs, le programme capable d'engranger les sources du code original et de produire le code fournissant les dérivées doit être capable de s'adapter aux spécificités du langage informatique dans lequel le

programme original est écrit. Il existe un nombre extrêmement important de logiciels capables de produire un code issu de la différenciation automatique [9] parmi lesquels on peut citer le logiciel ADIFOR [BISCHOF et al. 1992] adapté au langage *Fortran* ou TAPENADE [HASCOËT 2004 ; HASCOËT et ARAYA-POLO 2006] développé par l'INRIA capable de prendre en compte un code source en langage `C` ou `Fortran`.

3.2 Stratégie multiparamétrique et calcul des gradients

Dans le cadre du calcul d'assemblages employant la stratégie multiparamétrique et COFAST3D, il n'est à l'heure actuelle possible que d'employer une méthode de différences finies pour calculer les gradients. En effet, afin de profiter au mieux de la stratégie multiparamétrique et de ses performances en termes de réduction du temps de calcul pour des solutions proches, seule une approche non intrusive est adaptée. La complexité du comportement non-linéaire (en particulier le contact) présent dans les assemblages rend inenvisageable les méthodes directes et adjointes. Par ailleurs, la complexité du code COFAST3D du point de vue architecture rend la différenctiation automatique inemployable. En effet, COFAST3D est basé sur des routines en langage *Gibiane* et en langage *ESOPE*. Un certain nombre de procédures internes sont programmées en *Fortran* mais la multiplicité des langages et leur imbrication rendent impossible le déploiement d'un logiciel existant de différenciation automatique.

3.2.1 Problématique des différences finies appliquées à la stratégie multiparamétrique

Comme nous l'avons vu précédemment, la méthode de dérivation par différences finies, bien que d'un principe très simple, nécessite la détermination du pas. Cette détermination s'avère particulièrement complexe dans le cadre de l'emploi de la stratégie multiparamétrique. En effet, on peut identifier certaines spécificités rendant le calcul des gradients difficile :

- la stratégie multiparamétrique étant basée sur l'algorithme itératif LATIN, le calcul de plusieurs solutions « proches » (au sens des paramètres) peut entraîner l'obtention de réponses associées ne correspondant pas tout à fait au même niveau d'erreur. En effet, l'algorithme, suite à une initialisation, réalisera au minimum une itération pour obtenir la nouvelle solution. Les niveaux d'erreur atteints peuvent alors être différents pour les deux solutions. Ce facteur pouvant avoir un impact important sur la valeur de la réponse, lors d'une résolution pour calculer un point pour une différence finie, il est indispensable de vérifier le critère d'erreur à chaque itération (classiquement, ce critère est vérifié toutes les 5 itérations pour éviter un surcoût en temps de calcul).

- en raison de la méthode employée pour la prise en compte des interfaces (multiplicateurs de Lagrange), un résidu des grandeurs aux interfaces peut persister et

9. Voir par exemple `http://www.autodiff.org`

venir perturber la valeur réelle de la fonction. Ce niveau de perturbation n'atteignant pas des valeurs trop élevées, il perturbera les résultats majoritairement lorsque les grandeurs à mesurer seront d'un niveau faible. Par exemple, une interface faisant intervenir un phénomène de contact pourra présenter cette problématique lorsque le contact n'est pas présent (phase de décollement) : la valeur des efforts dans l'interface ne sera pas tout à fait nulle. Ainsi, des fluctuations de la réponse peuvent être relevées dans cette phase pour plusieurs jeux de paramètres et avoir un impact sur le gradients.

À ces deux points s'ajoute la problématique classique de détermination du pas telle qu'elle est présentée dans la partie 3.1.1 page 123.

3.2.2 Recherche du pas des différences finies

La détermination du pas est la difficulté principale lors de l'emploi de différences finies. En particulier dans le cas des assemblages de structures, il est important de garder à l'esprit que le coût des calculs ne permet en aucun cas de multiplier les résolutions pour évaluer les gradients. Par conséquent, il ne sera pas envisageable d'utiliser des différences finies d'ordre élevé.

Par ailleurs, les modifications de comportement d'un assemblage sur la plage de variation des paramètres considérés font intervenir des phénomènes variés ayant un impact non négligeable à la fois sur la convergence de l'algorithme et sur les résultats. On peut citer par exemple des cas d'assemblages ayant des interfaces siège de phénomène de glissement pour un certain jeu de paramètres et d'adhérence pour un autre jeu de paramètres ou des interfaces présentant alternativement contact et décollement en fonction des valeurs des jeux de paramètres.

■ **Nécessité d'une étude spécifique :** Cette problématique rendant la détermination automatique du pas optimal particulièrement ardue (en particulier la détermination du bruit numérique par l'intermédiaire de la constante ϵ_A), une étude permettant un choix du pas des différences finies a été envisagée. Cette étude allie l'obtention de résultats fortement convergés (critère d'arrêt basé sur l'erreur LATIN fixée à 10^{-5}), l'emploi de l'algorithme LATIN avec et sans stratégie multiparamétrique, et l'utilisation de schémas de différences finies d'ordres variés. Cette étude s'intègre bien entendu dans le contexte de changement de comportement de l'assemblage et des phénomènes qui interviennent en son sein.

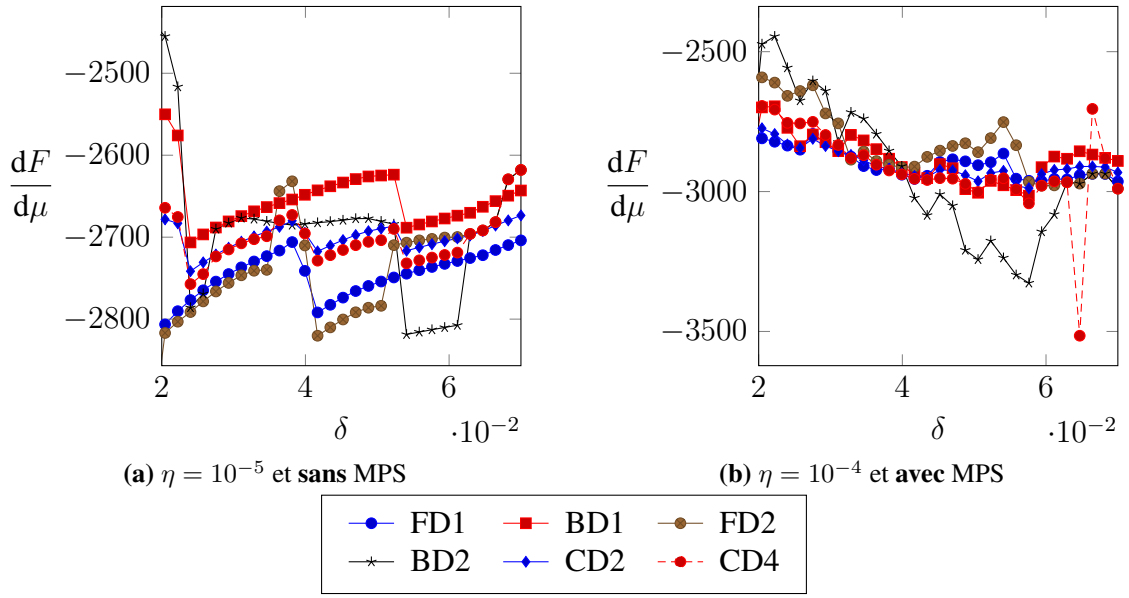
Remarque 2.15 : Des tentatives d'utilisation de l'algorithme FD permettant la détermination automatique du pas se sont avérées infructueuses. En particulier, la détermination de la constante ϵ_A en utilisant les outils développés par [MORÉ et WILD 2011] n'a pas abouti.

■ **Etude de pas :** Afin de proposer une valeur de pas fiable pour assurer l'obtention des gradients de la fonction objectif, les différents schémas de différences finies sont employés avec des pas de longueurs variables. La démarche est appliquée sur le cas-test

CSMA à 1 paramètre (cf. figure 2.2d page 98) et la dérivée de l'effort exercé par la sous-structure Ω_2 sur la paroi par rapport au coefficient de frottement est recherchée en deux valeurs de ce dernier. Afin d'étudier les dérivées obtenues par les différents schémas, les deux valeurs du coefficient de frottement retenu sont $\mu = 0.1$ et $\mu = 0.6$, c'est-à-dire deux valeurs du paramètre correspondant à deux comportements distincts de l'assemblage (avec et sans contact à la paroi). On pourra se référer à la figure 2.3d page 101 pour la valeur de l'effort en fonction du coefficient. Les figures 2.15 et 2.16 présentent les dérivées obtenues par les différents schémas de différences finies en employant un calcul fortement convergé (critère à 10^{-5}) avec et sans emploi de la stratégie multiparamétrique (désignée par le terme MPS - *MultiParametric Strategy*) et un critère d'arrêt fixé à 10^{-4}. Les différentes courbes sont tracées en fonction du paramètre adimensionné δ définissant le pas h des différences finies de la manière suivante :

$$h = \delta(b^u - b^l) \tag{2.32}$$

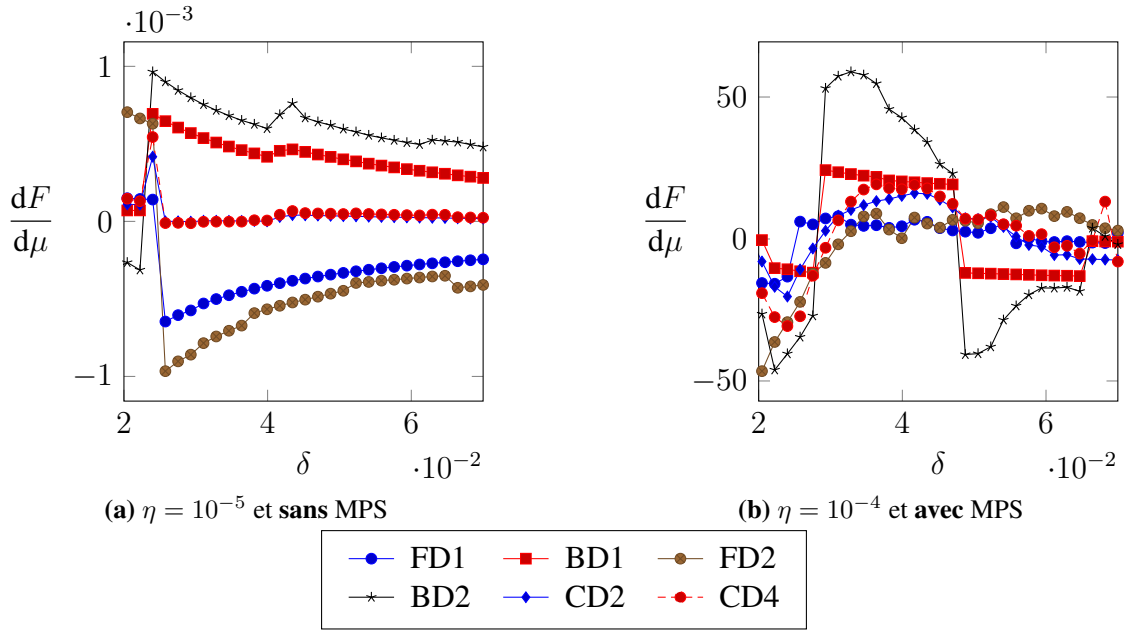
où b^u et b^l sont les bornes de l'espace des paramètres telles que $b^l \leq x \leq b^u$ (les valeurs de ces bornes sont données dans le tableau 2.1 page 99).



(a) $\eta = 10^{-5}$ et **sans** MPS **(b)** $\eta = 10^{-4}$ et **avec** MPS

FIGURE 2.15 – Gradients de l'effort à la paroi en fonction du coefficient de frottement à l'interface pour $\mu = 0.1$ (cas-test CSMA)

■ **Bilan :** L'étude proposée ici pour choisir le pas des différences finies s'avère complexe. En effet, les dérivées calculées sont particulièrement bruitées et rendent le choix du paramètre difficile. On constate cependant que les différents schémas proposés conduisent à des dérivées du même ordre de grandeur et ce pour les deux valeurs du critère d'arrêt de la méthode LATIN. Dans l'objectif de minimiser les temps de calcul et au vu



(a) $\eta = 10^{-5}$ et **sans** MPS **(b)** $\eta = 10^{-4}$ et **avec** MPS

FIGURE 2.16 – Gradients de l'effort à la paroi en fonction du coefficient de frottement à l'interface pour $\mu = 0.6$ (cas-test CSMA)

des écarts faibles constatés entre les différents schémas de différences finies, il a été décidé d'employer un schéma de différence finie d'ordre 1 de type FD1 ou BD1. Le pas, quant à lui, sera choisi par l'intermédiaire du paramètre δ et fixé à la valeur $\delta = 5\%$.

Remarque 2.16 : Dans l'objectif de compléter cette étude, une étude d'impact de la discrétisation du modèle éléments finis serait à envisager. Celle-ci devra s'atteler à étudier l'impact du type d'éléments et de leur densité. Par ailleurs, cette étude serait à renouveler sur l'ensemble des cas-tests dont on souhaiterait évaluer les gradients.

4 Illustrations des performances

La mise en œuvre complète (calcul des réponses et des gradients) de la stratégie multiparamétrique peut désormais être présentée sur trois cas-tests 2D simples. Ceux-ci comportent respectivement 2, 3 et 4 paramètres. Le premier n'est autre que le cas-test `CSMA` introduit dans la partie 2.3.2 (voir la géométrie en figure 2.2d page 98) dans sa version avec 2 paramètres (cf. page 2.3.6). On considère comme paramètre pour cet assemblage les deux coefficients de frottement aux interfaces $\Omega_1 - \Omega_2$ et $\Omega_2 - \Omega_3$. Le second assemblage n'est autre qu'une version dérivée du cas-test `CSMA` que l'on désignera sous le nom `CSMAb` et dont la géométrie est présentée par la figure 2.17.

Ces cas-tests se déclinent en deux versions : la première comporte les 4 paramètres tandis que la seconde ne comporte que 3 paramètres (2 coefficients de frottement tels que $\mu_1 = \mu_3$ et le jeu j_1).

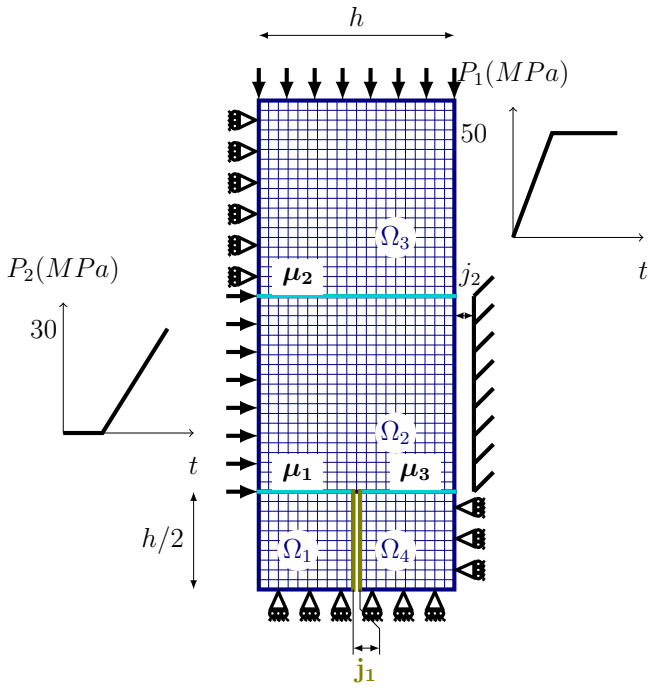


FIGURE 2.17 – Géométrie du cas-test CSMAb

Ce nouveau cas-test comporte 4 paramètres : 3 coefficients de frottement μ_1, μ_2 et μ_3 respectivement aux interfaces $\Omega_1 - \Omega_2$, $\Omega_2 - \Omega_3$ et $\Omega_2 - \Omega_4$ et un jeu j_1 à l'interface $\Omega_1 - \Omega_4$. Les coefficients de frottement prennent des valeurs entre 0 et 0.95 tandis que le jeu peut prendre des valeurs entre $-28\,\mu m$ et $48\,\mu m$.

L'ensemble des éléments choisis lors des parties précédentes est appliqué sur ces cas-tests : l'initialisation de l'algorithme LATIN est réalisée par la stratégie `para L2`, les jeux de paramètres sont échantillonnés par le biais d'une méthode LHS puis triés en utilisant la méthode `pb-sc` et finalement les gradients sont calculés à l'aide d'un schéma de différences finies avancées à l'ordre 1 avec $\delta = 5\%$.

Les résultats pour un tirage comportant 25 jeux de paramètres sont récapitulés dans le tableau 2.6. Les colonnes Δ montrent l'évolution des grandeurs lorsque l'on calcule les gradients.

Le tableau 2.6 illustre pleinement les performances associées à l'emploi de la stratégie multiparamétrique pour calculer les gradients : elle permet l'obtention des gradients en employant les différences finies sans augmenter de manière importante le coût des calculs. Si l'on considère le cas-test CSMAb à 3 paramètres, le temps de calcul total se trouve augmenté de seulement 57% alors que le nombre d'évaluations augmente de 300%. Par ailleurs, le gain de la stratégie croît également de manière significative pour aller, dans deux cas, jusqu'à doubler pour atteindre 8.31 et 6.18.

	CSMA			CSMAb (3 para.)			CSMAb (4 para.)		
Gradients	Non	Oui	Δ	Non	Oui	Δ	Non	Oui	Δ
Nb. d'appels au solveur	25	3×25	$+200\%$	25	4×25	$+300\%$	25	5×25	$+400\%$
Temps CPU total [s]	61.1	131.1	$+115\%$	140.5	221	$+57\%$	148.2	310.08	$+109\%$
Temps CPU moyen [s]	2.44	1.75	-28%	5.62	2.21	-61%	5.93	2.48	-58%
Temps CPU mini. [s]	0.17	0.1	-41%	0.7	0.13	-81%	0.88	0.13	-85%
Temps CPU maxi. [s]	9.15	9.18	–	19.53	18.36	–	18.24	15.33	–
Nb. itérations total.	2616	4698	$+80\%$	5956	6772	$+14\%$	6291	9222	$+47\%$
Nb. itérations moyen	104	63	-39%	238	68	-71%	252	74	-71%
Nb. itérations mini.	5	2	-60%	25	2	-92%	31	2	-94%
Nb. itérations maxi.	411	411	–	841	776	–	791	666	–
Gain_I	3.74	5.25	$+40\%$	3.47	8.31	$+139\%$	3.08	6.18	$+101\%$
Gain_M	3.74	5.25	$+40\%$	3.47	8.31	$+139\%$	3.08	6.18	$+101\%$

TABLE 2.6 – Caractéristiques des calculs avec et sans calcul des gradients pour les cas-tests CSMA et CSMAb avec 3 et 4 paramètres

5 Bilan sur l'emploi de la stratégie multiparamétrique pour l'optimisation

Ce chapitre a présenté l'emploi de la stratégie multiparamétrique pour résoudre un ensemble de problèmes d'assemblages similaires (où seuls quelques paramètres d'interface varient). Après une présentation des performances sur 4 exemples 2D faisant intervenir des paramètres et des comportement différents, des éléments permettant un choix pertinent du critère d'arrêt de l'algorithme LATIN ont été présentés. L'objectif était ici de proposer une valeur du critère d'arrêt permettant un bon compromis qualité/performance. Dans un second temps, une étude a été proposée pour choisir une stratégie d'initialisation pertinente, c'est-à-dire qui minimise le coût de calcul. C'est finalement une méthode de choix de la solution la plus proche dans l'espace des paramètres qui se distingue par les performances qu'elle engendre. Dans un troisième temps, la stratégie multiparamétrique est employée pour évaluer les gradients de la fonction objectif par différences finies. La principale problématique de cette méthode consiste en un choix de pas permettant l'obtention de gradients de qualité. Ce choix s'avère complexe, mais une valeur a pu être retenue. Finalement, la stratégie multiparamétrique est employée sur 3 exemples 2D pour calculer réponses et gradients. Son utilisation prend alors toute son importance. Elle assure en effet d'importantes réductions des coûts de calcul, en particulier lors du calcul des gradients.

Au bilan, la stratégie multiparamétrique apparaît comme un outil performant pour les études paramétrique et pour l'optimisation. Dans un contexte industriel et malgré ces réductions importantes de temps de calcul, ceux-ci peuvent rester importants. C'est pourquoi un élément complémentaire peut être envisagé pour assurer une aide à l'optimisation globale en réduisant encore les temps de calcul. C'est l'objet du chapitre suivant, dans lequel on propose de construire une approximation de la fonction objectif à partir de réponses et de gradients calculés en un nombre limité de points par la stratégie multiparamétrique.

Chapitre 3

Modèles de substitution et prise en compte des gradients

Ce chapitre s'articule autour de l'élaboration de métamodèles. Dans la continuité de la stratégie multiparamétrique capable de fournir les gradients de la fonction objectif, l'idée consiste ici à proposer des modèles de substitution capables de les prendre en compte. On se concentre sur des métamodèles de type krigeage et RBF dans leur version avec gradients. Après quelques éléments de théorie, leur capacités d'approximation seront étudiées sur des fonctions analytiques et lors de l'emploi de la stratégie multiparamétrique.

Sommaire

1 Introduction 134
2 Krigeage et RBF sans gradient 134
2.1 Krigeage 134
2.2 Fonctions de base radiales (RBF) 156
3 Modèles de substitution à gradients 160
3.1 Cokrigeage à gradients 160
3.2 RBF à gradients 171
3.3 Autres métamodèles à gradients 173
4 Comparaisons de métamodèles 173
4.1 Fonction d'une variable 174
4.2 Fonction de deux variables 178
4.3 Dimensions supérieures 184
4.4 Bilan 187
5 Métamodèles et stratégie multiparamétrique 188
5.1 Illustration sur un exemple simple 188
5.2 Application à des exemples comportant plus de variables de conception 193
6 Bilan de l'élaboration des modèles de substitution 198

1 Introduction

Comme nous l'avons vu dans le chapitre précédent, les gradients de la fonction objectif peuvent être obtenus avec un coût de calcul équivalent à celui nécessaire à l'obtention des réponses grâce à l'emploi de la stratégie multiparamétrique. Ces gradients constituent une information utile lors de la mise en œuvre de certains types d'algorithme d'optimisation tels que les algorithmes de descente et de Newton (cf. partie 2.3.2 du chapitre 1 page 19). Dans la pratique, une optimisation globale précise sur un modèle mécanique coûteux, comme ceux que l'on rencontre dans le cadre de l'optimisation d'assemblages, n'est pas viable en termes de temps de calcul. L'idée des travaux présentés dans le cadre de cette thèse est de substituer au modèle mécanique un **métamodèle** (ou modèle de substitution) capable de fournir à coût extrêmement faible des réponses approchées de la fonction objectif considérée. Les métamodèles classiques introduits dans la partie 4.5 du chapitre 1 page 75 sont simplement construits à partir de quelques réponses du modèle mécanique obtenues en quelques jeux de paramètres échantillonnés.

Dans cette partie, l'objectif est de déployer un modèle de substitution capable de prendre en compte non seulement les réponses de la fonction objectif, mais également ses gradients. Pour cela, on se concentre sur deux métamodèles classiquement employés : le krigeage et les fonctions de bases radiales.

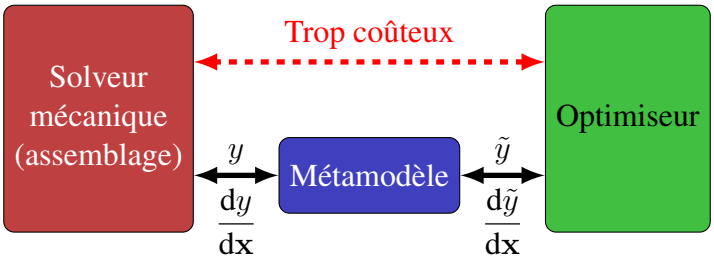


FIGURE 3.1 – Intégration d'un métamodèle pour approcher une réponse coûteuse

Ces deux modèles de substitution seront tout d'abord présentés dans leur version sans gradient, puis les extensions avec gradients seront introduites.

2 Krigeage et RBF sans gradient

2.1 Krigeage

Introduit dans les années 1950 par D.G. Krige en Afrique du Sud pour l'estimation de gisements miniers de mines d'or situées dans le massif du Witwatersrand, il propose des méthodes statistiques empiriques permettant de déterminer la répartition spatiale de minerais à partir de forages [KRIGE 1951a]. La méthode prend le nom de krigeage dans les années 1960 suite aux travaux de G. Matheron. Ce dernier mit en place un certain nombre de développements mathématiques de cette méthode [MATHERON 1962 ; MATHERON 1963a ;

MATHERON 1965]. Cette dernière fut par la suite largement déployée et reçut un certain nombre de compléments techniques dans le cadre de la géostatistique (voir par exemple [CRESSIE 1993 ; WACKERNAGEL 1995]). Le krigeage est également connu sous les noms de *méthode d'interpolation optimale* en météorologie et de *méthode d'interpolation de Gauss-Markov* en océanographie.

C'est finalement à la fin des années 1980 que les travaux de [SACKS, SCHILLER et WELCH 1989 ; SACKS, WELCH et al. 1989] proposent d'employer cette méthode dans le cadre de l'approximation de fonctions déterministes et de l'optimisation dans un contexte de simulations numériques. La méthode déployée par SACKS, WELCH et al. est connue sous le nom DACE (*Design and Analysis for Computer Experiments*). De nombreux compléments relatifs à cette approche peuvent être trouvés dans [SANTNER, B. J. WILLIAMS et NOTZ 2003]. Par ailleurs, des développements dans le cadre du formalisme Bayésien ont donné lieu à l'apparition des modèles d'interpolation et de régression par Processus Gaussien [1] [CURRIN et al. 1991 ; NEAL 1998 ; M. KENNEDY et O'HAGAN 2000 ; C. E. RASMUSSEN et C. K. I. WILLIAMS 2006].

Le krigeage est aujourd'hui très utilisé en raison de la qualité de l'approximation qu'il peut fournir et de sa robustesse. Par ailleurs, il fournit une estimation de l'erreur des moindres carrés du processus qu'il estime. Cette information s'avère particulièrement utile dans un cadre de propagation d'incertitudes par exemple ou, comme on le verra par la suite, dans l'objectif d'enrichir un métamodèle existant de manière intelligente.

2.1.1 Élaboration du krigeage

Dans le contexte de ces travaux, on vise à construire une approximation d'une fonction supposée déterministe à partir de ces valeurs en quelques points échantillonnés. Pour cela, le krigeage propose de se baser sur un modèle probabiliste et considère que la véritable fonction y est une réalisation d'un processus Gaussien stationnaire Y. Ce dernier se formule de la manière suivante :

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + Z(\mathbf{x}) \tag{3.1}$$

où $\mu(\mathbf{x})$ est une quantité déterministe et $Z(\mathbf{x})$ une quantité aléatoire. Plus précisément, Z est un processus gaussien à moyenne nulle ($\mathbb{E}[Z(\mathbf{x})] = 0$) et de fonction de covariance k. Ainsi, l'espérance et la variance de $Y(\mathbf{x})$ s'écrivent :

$$\mathbb{E}[Y(\mathbf{x})] = \mathbb{E}[\mu(\mathbf{x}) + Z(\mathbf{x})] = \mu(\mathbf{x}) \tag{3.2}$$

$$\sigma_Y^2 = \text{Var}[Y(\mathbf{x})] = \text{Var}[\mu(\mathbf{x}) + Z(\mathbf{x})] = \text{Var}[Z(\mathbf{x})] = \sigma_Z^2 \tag{3.3}$$

Par ailleurs, la relation de covariance se formule de la manière suivante :

$$k\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\big) = \text{cov}\big[Y\big(\mathbf{x}^{(i)}\big), Y\big(\mathbf{x}^{(j)}\big)\big] = \text{cov}\big[Z\big(\mathbf{x}^{(i)}\big), Z\big(\mathbf{x}^{(j)}\big)\big] \tag{3.4}$$

1. De nombreuses logiciels et références concernant les processus gaussiens sont disponibles sur le site `http://www.gaussianprocess.org`.

Remarque 3.1 : Dans le cadre de la géostatistique et des travaux de Matheron, le modèle proposé par l'équation 3.1 est celui employé pour qualifier une *variable régionalisée* [MATHERON 1970].

Remarque 3.2 : Pour des compléments sur le formalisme utilisé pour construire le krigeage et les processus gaussiens, se référer à [C. E. RASMUSSEN et C. K. I. WILLIAMS 2006].

2.1.2 Principe de construction

Le krigeage consiste à chercher à déterminer un estimateur $\hat{Y}\big(\mathbf{x}^{(0)}\big)$ de la valeur en un site $\mathbf{x}^{(0)} \in \mathcal{D}$ du processus aléatoire $Y(\mathbf{x})$ ($\mathbf{x} \in \mathcal{D}$). Connaissant un échantillon E_a constitué de n_s observations y_i calculées en un ensemble de sites $\mathbf{x}^{(i)}$ obtenus grâce à un plan d'expérience, l'estimateur est cherché sous la forme d'une combinaison linéaire telle que celle présentée par l'équation 3.5.

$$\hat{Y}\big(\mathbf{x}^{(0)}\big) = \sum_{i=1}^{n_s} \lambda_i\big(\mathbf{x}^{(0)}\big) Y_i = \boldsymbol{\lambda}\big(\mathbf{x}^{(0)}\big)^T \mathbf{Y}_{n_s} \tag{3.5}$$

où $\lambda_i\big(\mathbf{x}^{(0)}\big) \in \mathbb{R}$ sont les coefficients du krigeage et constituent les inconnues du problème. Les notations adoptées ici sont :

$$\boldsymbol{\lambda}\big(\mathbf{x}^{(0)}\big) = \big[\lambda_1\big(\mathbf{x}^{(0)}\big) \quad \lambda_2\big(\mathbf{x}^{(0)}\big) \quad \dots \quad \lambda_{n_s}\big(\mathbf{x}^{(0)}\big)\big]^T$$
$$\mathbf{Y}_{n_s} = \big[Y_1 \quad Y_2 \quad \dots \quad Y_{n_s}\big]^T \text{ (où } \forall i \in [\![1, n_s]\!],\ Y_i = Y\big(\mathbf{x}^{(i)}\big)\text{)}$$

Remarque 3.3 : L'estimateur est parfois cherché sous la forme :

$$\hat{Y}\big(\mathbf{x}^{(0)}\big) = a + \sum_{i=1}^{n_s} \lambda_i\big(\mathbf{x}^{(0)}\big) Y_i$$

On montre classiquement que pour les types de krigeage considérés dans ces travaux, l'existence du moment statistique d'ordre 2 de l'erreur de prévision (équation 3.6) entraîne $a = 0$.

Pour des compléments sur ce point et sur la théorie du krigeage dans le cadre géostatistique se référer à [CHAUVET 1999 ; BAILLARGEON 2005].

Remarque 3.4 : Pour des raisons de lisibilité, la notation λ sera employée pour désigner le coefficient $\lambda\big(\mathbf{x}^{(0)}\big)$. Il faut garder à l'esprit que ces **coefficients restent dépendants de** $\mathbf{x}^{(0)}$, le point où l'on cherche l'estimateur.

La détermination des inconnues est réalisée en s'assurant que l'estimateur est **linéaire**, **sans biais** et à **variance minimale** (on parle de BLUP - *Best Linear Unbiased Predictor*) comme le rappelle la définition suivante :

Définition 3.1 (Best Linear Unbiased Predictor) **:** Un estimateur $\hat{f}(x)$ d'une variable aléatoire $f(x)$ est le meilleur estimateur linéaire non biaisé si :

- il est sans biais : $\mathbb{E}\Big[f(x) - \hat{f}(x)\Big] = 0$
- il minimise la variance de l'erreur de prévision (il minimise l'erreur quadratique moyenne) $\text{Var}\Big[f(x) - \hat{f}(x)\Big] = \mathbb{E}\Big[\Big(f(x) - \hat{f}(x)\Big)^2\Big]$

Ainsi, la condition de non-biais (équation 3.7) se formule à partir de l'erreur de prévision $\varepsilon\big(\mathbf{x}^{(0)}\big)$:

$$\begin{aligned}
\varepsilon\big(\mathbf{x}^{(0)}\big) &= \hat{Y}\big(\mathbf{x}^{(0)}\big) - Y\big(\mathbf{x}^{(0)}\big) \\
&= \sum_{i=1}^{n_s} \lambda_i Y_i - Y_0 \\
&= \sum_{i=1}^{n_s} \lambda_i \left(\mu\big(\mathbf{x}^{(i)}\big) + Z\big(\mathbf{x}^{(i)}\big)\right) - \left(\mu\big(\mathbf{x}^{(0)}\big) + Z\big(\mathbf{x}^{(0)}\big)\right) \\
&= \underbrace{\sum_{i=1}^{n_s} \lambda_i \mu\big(\mathbf{x}^{(i)}\big) - \mu\big(\mathbf{x}^{(0)}\big)}_{\text{Déterministe}} + \underbrace{\sum_{i=1}^{n_s} \lambda_i Z\big(\mathbf{x}^{(i)}\big) - Z\big(\mathbf{x}^{(0)}\big)}_{\text{Aléatoire}}
\end{aligned} \tag{3.6}$$

On emploiera par la suite la notation Z_i pour désigner la grandeur $Z\big(\mathbf{x}^{(i)}\big)$ ($\forall i \in [\![0, n_s]\!]$). La condition de non-biais s'écrit alors

$$\begin{aligned}
\mathbb{E}\big[\varepsilon\big(\mathbf{x}^{(0)}\big)\big] &= 0 \\
\Leftrightarrow \mathbb{E}\left[\sum_{i=1}^{n_s} \lambda_i Y_i - Y_0\right] &= 0 \\
\Leftrightarrow \sum_{i=1}^{n_s} \lambda_i \mu\big(\mathbf{x}^{(i)}\big) - \mu\big(\mathbf{x}^{(0)}\big) &= 0
\end{aligned} \tag{3.7}$$

Cette condition de non-biais engendre des contraintes portant sur les coefficients du krigeage λ. Il reste maintenant à calculer la variance de krigeage qui s'exprime :

$$\begin{aligned}
\text{Var}\left[\varepsilon\left(\mathbf{x}^{(0)}\right)\right] &= \text{Var}\left[\sum_{i=1}^{n_s}\lambda_i Y_i - Y_0\right] \\
&= \text{Var}\left[\sum_{i=1}^{n_s}\lambda_i Z_i - Z_0\right] \text{ car } Y(\mathbf{x}) = \mu(\mathbf{x}) + Z(\mathbf{x}) \text{ et } \text{Var}[\mu(\mathbf{x})] = 0 \\
&= \text{Var}\left[\sum_{i=1}^{n_s}\lambda_i Z_i\right] - 2\text{cov}\left[\sum_{i=1}^{n_s}\lambda_i Z_i, Z_0\right] + \text{Var}[Z_0] \\
&= \sum_{i=1}^{n_s}\sum_{i=1}^{n_s}\lambda_i\lambda_j \text{cov}[Z_i, Z_j] - 2\sum_{i=1}^{n_s}\lambda_i \text{cov}[Z_i, Z_0] + \text{Var}[Z_0] \\
&= \boldsymbol{\lambda}^T\mathbf{C}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}^T\mathbf{c}_0 + \sigma_Z^2 \qquad (3.8)
\end{aligned}$$

où $\mathbf{C}$ est la matrice de covariance, $\mathbf{c}_0$ est le vecteur des covariances entre le site $\mathbf{x}^{(0)}$ et les sites $\mathbf{x}^{(i)}$ et $\sigma_Z = \text{Var}[Z_0]^2 = \text{Var}[Y_0] = \sigma_Y^2$. Finalement, construire un modèle de krigeage, c'est-à-dire un estimateur $\hat{Y}$ au point $\mathbf{x}^{(0)}$ revient à résoudre le problème suivant :

Problème 3.1 (Modèle de krigeage)
Trouver $\boldsymbol{\lambda} \in \mathbb{R}^{n_s}$ minimisant

$$\boldsymbol{\lambda}^T\mathbf{C}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}^T\mathbf{c}_0 + \sigma_Z^2 \qquad (3.9)$$

sous la contrainte $\sum_{i=1}^{n_s}\lambda_i\mu\left(\mathbf{x}^{(i)}\right) - \mu\left(\mathbf{x}^{(0)}\right) = 0$

Remarque 3.5 : Dans le cadre de la géostatistique, les différentes étapes de la construction du krigeage sont identifiées sous les termes [CHAUVET 1999] :

- Contrainte de **linéarité** : l'estimateur est cherché sous la forme d'une combinaison linéaire des données disponibles.
- Contrainte d'**autorisation** : les moments statistiques d'ordre 1 et 2 de l'erreur d'estimation existent. Autrement dit, l'erreur d'estimation est autorisée.
- Contrainte d'**universalité** : elle consiste à dire que l'erreur d'estimation est d'espérance nulle.
- Contrainte d'**optimalité** : elle consiste à imposer à l'erreur d'estimation d'être à variance minimale.

2.1.3 Différents types de krigeage

On distingue classiquement 3 types de krigeage : le krigeage simple (KS) où la quantité déterministe $\mu\left(\mathbf{x}^{(0)}\right)$ est considérée **constante**, **connue** et valant m, le krigeage ordinaire

(KO) où $\mu\big(\mathbf{x}^{(0)}\big)$ est considérée **constante**, **inconnue** et valant β et le krigeage universel (KU) où $\mu\big(\mathbf{x}^{(0)}\big)$ est considérée comme une combinaison linéaire de p fonctions de base $\{f_1, f_2, \ldots, f_p\}$ telle que $\mu\big(\mathbf{x}^{(0)}\big) = \sum_{i=1}^{p} \beta_i f_i\big(\mathbf{x}^{(0)}\big)$. La différence entre ces 3 types de krigeage se joue finalement au niveau de la partie déterministe (la tendance) du modèle probabiliste. Ces différences peuvent s'avérer utiles lorsque l'utilisateur possède une connaissance *a priori* de l'évolution de la fonction. Il peut alors chercher à prédisposer le modèle à tendre vers cette évolution attendue de la réponse.

■ **Krigeage Simple** Dans le cas du krigeage simple, la moyenne du processus étant connue et constante, on peut parfaitement se ramener à un processus gaussien à moyenne nulle par la transformation $Y(\mathbf{x}) - m \rightarrow Y(\mathbf{x})$. Le processus étudié s'écrit alors simplement

$$Y\left(\mathbf{x}\right) = Z(\mathbf{x}) \tag{3.10}$$

La contrainte de non-biais se trouve alors automatiquement vérifiée ($\forall i \in [\![0, n_s]\!],\ \mu(\mathbf{x}^{(i)}) = 0$). Les n_s coefficients λ_i minimisant la variance de l'erreur d'estimation peuvent alors être obtenus en résolvant

$$\begin{aligned} \frac{\mathrm{dVar}\big[\varepsilon(\mathbf{x}^{(0)})\big]}{\mathrm{d}\boldsymbol{\lambda}} &= \mathbf{0} \\ \mathbf{C}\boldsymbol{\lambda} - \mathbf{c}_0 &= \mathbf{0} \end{aligned} \tag{3.11}$$

Si $\mathbf{C}$ est inversible, on obtient alors $\hat{\boldsymbol{\lambda}} = \mathbf{C}^{-1}\mathbf{c}_0$. L'estimateur du krigeage simple s'écrit alors dans le cas d'un processus à moyenne nulle $\hat{Y}\big(\mathbf{x}^{(0)}\big) = \mathbf{c}_0^T\mathbf{C}^{-1}\mathbf{Y}_{n_s}$. Dans le cas général, on obtient :

$$\forall \mathbf{x}^{(0)} \in \mathcal{D},\ \hat{Y}_{KS}\big(\mathbf{x}^{(0)}\big) = m + \mathbf{c}_0^T\mathbf{C}^{-1}\left(\mathbf{Y}_{n_s} - m\mathbf{1}_{n_s}\right) \tag{3.12}$$

L'approximation de la réponse fournie par le krigeage simple vaut alors :

$$\boxed{\forall \mathbf{x}^{(0)} \in \mathcal{D},\ \tilde{y}_{KS}\big(\mathbf{x}^{(0)}\big) = \mathbb{E}\Big[\hat{Y}_{KS}\big(\mathbf{x}^{(0)}\big)\Big] = m + \mathbf{c}_0^T\mathbf{C}^{-1}\left(\mathbf{y}_{n_s} - m\mathbf{1}_{n_s}\right)} \tag{3.13}$$

où $\mathbf{y}_{n_s} = \begin{bmatrix} y_1 & \ldots & y_{n_s} \end{bmatrix}^T$ (avec $\forall i \in [\![1, n_s]\!],\ y_i = y\big(\mathbf{x}^{(i)}\big)$).

La variance du krigeage se calcule de la manière suivante :

$$\boxed{\forall \mathbf{x}^{(0)} \in \mathcal{D},\ s^2_{KS}\big(\mathbf{x}^{(0)}\big) = \mathrm{Var}\big[\varepsilon\left(\mathbf{x}^{(0)}\right)\big] = \sigma_Z^2 - \mathbf{c}_0^T\mathbf{C}^{-1}\mathbf{c}_0} \tag{3.14}$$

■ **Krigeage Ordinaire** Dans le cas du krigeage ordinaire, le processus gaussien est de la forme

$$Y\left(\mathbf{x}\right) = \beta + Z(\mathbf{x}) \tag{3.15}$$

où $\beta \in \mathbb{R}$. La contrainte de non-biais conduit alors à la relation :

$$\sum_{i=1}^{n_s} \lambda_i = 1 \text{ soit } \boldsymbol{\lambda}^T\mathbf{1}_{n_s} = 1 \tag{3.16}$$

Le problème 3.1 devient alors :

Problème 3.2 (Krigeage Odinaire)
Trouver $\boldsymbol{\lambda} \in \mathbb{R}^{n_s}$ minimisant

$$\boldsymbol{\lambda}^T \mathbf{C} \boldsymbol{\lambda} - 2\boldsymbol{\lambda}^T \mathbf{c}_0 + \sigma_Z^2 \tag{3.17}$$

sous la contrainte $\boldsymbol{\lambda}^T \mathbf{1}_{n_s} = 1$

En ajoutant un multiplicateur de Lagrange ξ associé à la contrainte du problème précédent, il est possible de formuler le Lagrangien suivant :

$$\mathcal{L}_{KO}(\boldsymbol{\lambda}, \xi) = \boldsymbol{\lambda}^T \mathbf{C} \boldsymbol{\lambda} - 2\boldsymbol{\lambda}^T \mathbf{c}_0 + \sigma_Z^2 + 2\xi \left(\boldsymbol{\lambda}^T \mathbf{1}_{n_s} - 1 \right) \tag{3.18}$$

La matrice $\mathbf{C}$ étant définie positive, l'unique point critique de $\mathcal{L}_{KO}$ et solution du problème 3.2 peut être déterminé tel qu'il annule les dérivées du Lagrangien :

$$\left| \begin{array}{l} \dfrac{\partial \mathcal{L}_{KO}}{\partial \boldsymbol{\lambda}} (\boldsymbol{\lambda}, \xi) = \mathbf{0} \Longrightarrow \mathbf{C}\boldsymbol{\lambda} - \mathbf{c}_0 + \xi \mathbf{1}_{n_s} = \mathbf{0} \\ \dfrac{\partial \mathcal{L}_{KO}}{\partial \xi} (\boldsymbol{\lambda}, \xi) = 0 \Longrightarrow \boldsymbol{\lambda}^T \mathbf{1}_{n_s} = 1 \end{array} \right. \tag{3.19}$$

Ce qui conduit au système matriciel du krigeage ordinaire comportant $n_s + 1$ équations linéaires :

$$\begin{bmatrix} \mathbf{C} & \mathbf{1}_{n_s} \\ \mathbf{1}_{n_s}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \xi \end{bmatrix} = \begin{bmatrix} \mathbf{c}_0 \\ 1 \end{bmatrix} \tag{3.20}$$

Finalement, la résolution de ce problème donne

$$\left| \begin{array}{l} \hat{\xi} = \left(\mathbf{1}_{n_s}^T \mathbf{C}^{-1} \mathbf{1}_{n_s} \right)^{-1} \left(\mathbf{1}_{n_s}^T \mathbf{C}^{-1} \mathbf{c}_0 - 1 \right) \\ \hat{\boldsymbol{\lambda}} = \mathbf{C}^{-1} \left(\mathbf{c}_0 + \dfrac{1 - \mathbf{1}_{n_s}^T \mathbf{C}^{-1} \mathbf{c}_0}{\mathbf{1}_{n_s}^T \mathbf{C}^{-1} \mathbf{1}_{n_s}} \mathbf{1}_{n_s} \right) \end{array} \right. \tag{3.21}$$

L'estimateur du krigeage ordinaire s'écrit alors

$$\forall \mathbf{x}^{(0)} \in \mathcal{D},\ \hat{Y}_{KO}\left(\mathbf{x}^{(0)}\right) = \left(\mathbf{c}_0 + \frac{1 - \mathbf{1}_{n_s}^T \mathbf{C}^{-1} \mathbf{c}_0}{\mathbf{1}_{n_s}^T \mathbf{C}^{-1} \mathbf{1}_{n_s}} \mathbf{1}_{n_s} \right)^T \mathbf{C}^{-1} \mathbf{Y}_{n_s} \tag{3.22}$$

On obtient l'approximation de la fonction ainsi que la variance fournie par le krigeage ordinaire :

$$\boxed{\forall \mathbf{x}^{(0)} \in \mathcal{D},\ \tilde{y}_{KO}\left(\mathbf{x}^{(0)}\right) = \left(\mathbf{c}_0 + \frac{1 - \mathbf{1}_{n_s}^T \mathbf{C}^{-1} \mathbf{c}_0}{\mathbf{1}_{n_s}^T \mathbf{C}^{-1} \mathbf{1}_{n_s}} \mathbf{1}_{n_s} \right)^T \mathbf{C}^{-1} \mathbf{y}_{n_s}} \tag{3.23}$$

$$\boxed{\begin{aligned} \forall \mathbf{x}^{(0)} \in \mathcal{D},\ s_{KO}^2\left(\mathbf{x}^{(0)}\right) &= \mathrm{Var}\left[\varepsilon\left(\mathbf{x}^{(0)}\right)\right] \\ &= \sigma_Z^2 - \mathbf{c}_0^T \mathbf{C}^{-1} \mathbf{c}_0 + \mathbf{u}_0^T \left(\mathbf{1}_{n_s}^T \mathbf{C}^{-1} \mathbf{1}_{n_s} \right)^{-1} \mathbf{u}_0 \\ \text{avec } \mathbf{u}_0 = \mathbf{u}\left(\mathbf{x}^{(0)}\right) &= \mathbf{1}_{n_s}^T \mathbf{C}^{-1} \mathbf{c}_0 - 1 \end{aligned}} \tag{3.24}$$

Remarque 3.6 : Une forme plus adaptée de la variance au cas où l'inverse de la matrice $\begin{bmatrix} \mathbf{C} & \mathbf{1}_{n_s} \\ \mathbf{1}_{n_s}^T & 0 \end{bmatrix}$ est connu s'écrit :

$$\begin{aligned} \forall \mathbf{x}^{(0)} \in \mathcal{D},\ s_{KO}^2\big(\mathbf{x}^{(0)}\big) &= \mathrm{Var}\big[\varepsilon\left(\mathbf{x}^{(0)}\right)\big] \\ &= \sigma_Z^2 - \begin{bmatrix} \mathbf{c}_0 \\ 1 \end{bmatrix}^T \begin{bmatrix} \mathbf{C} & \mathbf{1}_{n_s} \\ \mathbf{1}_{n_s}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{c}_0 \\ 1 \end{bmatrix} \end{aligned} \tag{3.25}$$

■ **Krigeage Universel** Dans le cas du krigeage universel, le processus gaussien est de la forme

$$Y\left(\mathbf{x}\right) = \sum_{i=1}^{p} \beta_i f_i\big(\mathbf{x}^{(0)}\big) + Z(\mathbf{x}) \tag{3.26}$$

où $\forall i \in [\![1, p]\!],\ \beta_i \in \mathbb{R}$. La contrainte de non-biais conduit alors à la relation :

$$\begin{aligned} \sum_{i=1}^{n_s} \lambda_i \sum_{j=1}^{p} \beta_j f_j\big(\mathbf{x}^{(i)}\big) &= \sum_{j=1}^{p} \beta_j f_j\big(\mathbf{x}^{(0)}\big) \\ \boldsymbol{\lambda}^T \mathbf{F} \boldsymbol{\beta} &= \mathbf{f}_0^T \boldsymbol{\beta} \end{aligned} \tag{3.27}$$

où $\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & \beta_2 & \dots & \beta_p \end{bmatrix}^T$, $\mathbf{f}_0 = \begin{bmatrix} f_1\big(\mathbf{x}^{(0)}\big) & f_2\big(\mathbf{x}^{(0)}\big) & \dots & f_p\big(\mathbf{x}^{(0)}\big) \end{bmatrix}^T$ et $\mathbf{F}$ est la matrice $p \times n_s$:

$$\mathbf{F} = \begin{bmatrix} f_1\big(\mathbf{x}^{(1)}\big) & f_2\big(\mathbf{x}^{(1)}\big) & \dots & f_p\big(\mathbf{x}^{(1)}\big) \\ f_1\big(\mathbf{x}^{(2)}\big) & \ddots & & \\ \vdots & & & \\ f_1\big(\mathbf{x}^{(n_s)}\big) & \dots & \dots & f_p\big(\mathbf{x}^{(n_s)}\big) \end{bmatrix}$$

La quantité déterministe étant *a priori* non nulle, la condition de non-biais s'écrit finalement

$$\mathbf{F}^T \boldsymbol{\lambda} = \mathbf{f}_0 \tag{3.28}$$

Le problème 3.1 devient alors :

Problème 3.3 (Krigeage Universel)
Trouver $\boldsymbol{\lambda} \in \mathbb{R}^{n_s}$ minimisant

$$\boldsymbol{\lambda}^T \mathbf{C} \boldsymbol{\lambda} - 2\boldsymbol{\lambda}^T \mathbf{c}_0 + \sigma_Z^2 \tag{3.29}$$

sous la contrainte $\mathbf{F}^T \boldsymbol{\lambda} = \mathbf{f}_0$

Avec une approche similaire à celle employée pour le krigeage ordinaire, en ajoutant p multiplicateurs de Lagrange ξ_i associés à la contrainte du problème précédent, il est possible de formuler le Lagrangien suivant en posant $\boldsymbol{\xi} = \begin{bmatrix} \xi_1 & \xi_2 & \dots \xi_p \end{bmatrix}^T$

$$\mathcal{L}_{KU}(\boldsymbol{\lambda}, \boldsymbol{\xi}) = \boldsymbol{\lambda}^T \mathbf{C} \boldsymbol{\lambda} - 2\boldsymbol{\lambda}^T \mathbf{c}_0 + \sigma_Z^2 + 2\boldsymbol{\xi}^T \left(\mathbf{F}^T \boldsymbol{\lambda} - \mathbf{f}_0\right) \tag{3.30}$$

L'annulation des dérivées partielles du Lagrangien par rapport à $\boldsymbol{\lambda}$ et $\boldsymbol{\xi}$ conduit à :

$$\left| \begin{array}{l} \dfrac{\partial \mathcal{L}_{KU}}{\partial \boldsymbol{\lambda}}(\boldsymbol{\lambda}, \boldsymbol{\xi}) = \mathbf{0} \Longrightarrow \mathbf{C}\boldsymbol{\lambda} - \mathbf{c}_0 + \mathbf{F}\boldsymbol{\xi} = \mathbf{0} \\ \dfrac{\partial \mathcal{L}_{KU}}{\partial \boldsymbol{\xi}}(\boldsymbol{\lambda}, \boldsymbol{\xi}) = 0 \Longrightarrow \mathbf{F}^T\boldsymbol{\lambda} = \mathbf{f}_0 \end{array} \right. \tag{3.31}$$

Ce qui conduit au système matriciel du krigeage universel comportant $n_s + p$ équations linéaires :

$$\begin{bmatrix} \mathbf{C} & \mathbf{F} \\ \mathbf{F}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\xi} \end{bmatrix} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{f}_0 \end{bmatrix} \tag{3.32}$$

Finalement, la résolution de ce problème donne

$$\left| \begin{array}{l} \hat{\boldsymbol{\xi}} = \left(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F}\right)^{-1}\left(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{c}_0 - \mathbf{f}_0\right) \\ \hat{\boldsymbol{\lambda}} = \mathbf{C}^{-1}\left(\mathbf{c}_0 + \mathbf{F}\left(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F}\right)^{-1}\left(\mathbf{f}_0 - \mathbf{F}^T\mathbf{C}^{-1}\mathbf{c}_0\right)\right) \end{array} \right. \tag{3.33}$$

L'estimateur du krigeage universel s'écrit alors

$\forall \mathbf{x}^{(0)} \in \mathcal{D}$,

$$\hat{Y}_{KU}\left(\mathbf{x}^{(0)}\right) = \left(\mathbf{c}_0 + \mathbf{F}\left(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F}\right)^{-1}\left(\mathbf{f}_0 - \mathbf{F}^T\mathbf{C}^{-1}\mathbf{c}_0\right)\right)^T \mathbf{C}^{-1}\mathbf{Y}_{n_s} \tag{3.34}$$

On en déduit l'approximation de la fonction par krigeage universel ainsi que la variance :

$$\boxed{\begin{array}{l} \forall \mathbf{x}^{(0)} \in \mathcal{D}, \\ \qquad \tilde{y}_{KU}\left(\mathbf{x}^{(0)}\right) = \left(\mathbf{c}_0 + \mathbf{F}\left(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F}\right)^{-1}\left(\mathbf{f}_0 - \mathbf{F}^T\mathbf{C}^{-1}\mathbf{c}_0\right)\right)^T \mathbf{C}^{-1}\mathbf{y}_{n_s} \end{array}} \tag{3.35}$$

$$\boxed{\begin{array}{l} \forall \mathbf{x}^{(0)} \in \mathcal{D},\ s^2_{KU}\left(\mathbf{x}^{(0)}\right) = \mathrm{Var}\left[\varepsilon\left(\mathbf{x}^{(0)}\right)\right] \\ \qquad\qquad = \sigma_Z^2 - \mathbf{c}_0^T\mathbf{C}^{-1}\mathbf{c}_0 + \mathbf{u}_0^T\left(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F}\right)^{-1}\mathbf{u}_0 \\ \qquad\qquad \text{avec } \mathbf{u}_0 = \mathbf{u}\left(\mathbf{x}^{(0)}\right) = \mathbf{F}^T\mathbf{C}^{-1}\mathbf{c}_0 - \mathbf{f}_0 \end{array}} \tag{3.36}$$

Remarque 3.7 : À l'instar du krigeage ordinaire, une forme plus adaptée de la variance au cas où l'inverse de la matrice $\begin{bmatrix} \mathbf{C} & \mathbf{F} \\ \mathbf{F}^T & 0 \end{bmatrix}$ est connu s'écrit :

$$\begin{array}{l} \forall \mathbf{x}^{(0)} \in \mathcal{D},\ s^2_{KU}\left(\mathbf{x}^{(0)}\right) = \mathrm{Var}\left[\varepsilon\left(\mathbf{x}^{(0)}\right)\right] \\ \qquad\qquad = \sigma_Y^2 - \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{f}_0 \end{bmatrix}^T \begin{bmatrix} \mathbf{C} & \mathbf{F} \\ \mathbf{F}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{f}_0 \end{bmatrix} \end{array} \tag{3.37}$$

Remarque 3.8 : On constate que dans le cas où l'on considère une seule fonction de base valant 1 ($f_1(\mathbf{x}) = 1$), on obtient $\mathbf{f}_0 = 1$ et $\mathbf{F} = \mathbf{1}_{n_s}$. On retrouve alors le modèle de krigeage ordinaire.

■ **Récapitulatif :** Suite à la présentation des estimateurs des 3 types de krigeage, il est possible d'identifier pour chacun d'eux une partie traduisant une évolution globale de la fonction et une correspondant à des corrections autour de celle-ci. Ces grandeurs ne sont pas à confondre avec les parties déterministe et aléatoire du processus gaussien retenues pour le krigeage. Le tableau suivant récapitule ces deux parties identifiées respectivement $\tilde{y}_g\big(\mathbf{x}^{(0)}\big)$ et $\tilde{y}_a\big(\mathbf{x}^{(0)}\big)$ définies telles que $\tilde{y}\big(\mathbf{x}^{(0)}\big) = \tilde{y}_g\big(\mathbf{x}^{(0)}\big) + \tilde{y}_a\big(\mathbf{x}^{(0)}\big)$.

	$\tilde{y}_g\big(\mathbf{x}^{(0)}\big)$	$\tilde{y}_a\big(\mathbf{x}^{(0)}\big)$	$\hat{\boldsymbol{\beta}}$
KS	m	$\mathbf{c}_0^T\mathbf{C}^{-1}\left(\mathbf{y}_{n_s} - m\mathbf{1}_{n_s}\right)$	–
KO	$\hat{\beta}$	$\mathbf{c}_0^T\mathbf{C}^{-1}\left(\mathbf{y}_{n_s} - \hat{\beta}\mathbf{1}_{n_s}\right)$	$\left(\mathbf{1}_{n_s}^T\mathbf{C}^{-1}\mathbf{1}_{n_s}\right)^{-1}\mathbf{1}_{n_s}^T\mathbf{C}^{-1}\mathbf{y}_{n_s}$
KU	$\mathbf{f}_0^T\hat{\boldsymbol{\beta}}$	$\mathbf{c}_0^T\mathbf{C}^{-1}\left(\mathbf{y}_{n_s} - \mathbf{F}\hat{\boldsymbol{\beta}}\right)$	$\left(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{C}^{-1}\mathbf{y}_{n_s}$

TABLE 3.1 – Composantes des approximations fournies par les 3 modèles de krigeage

L'évaluation de l'approximation $\tilde{y}_\bullet$ et de la variance $s_\bullet^2$ nécessite, une fois le modèle construit, uniquement l'évaluation des vecteurs $\mathbf{f}_0$ et $\mathbf{c}_0$ de taille respectivement p et n_s (dans le cas des krigeages simple et ordinaire, seul le vecteur $\mathbf{c}_0$ est à évaluer pour chaque site $\mathbf{x}^{(0)}$) tels que :

$$\mathbf{f}_0 = \begin{bmatrix} f_1\big(\mathbf{x}^{(0)}\big) & f_2\big(\mathbf{x}^{(0)}\big) & \dots & f_p\big(\mathbf{x}^{(0)}\big) \end{bmatrix}$$
$$\mathbf{c}_0 = \begin{bmatrix} \mathrm{cov}[Z_0, Z_1] & \mathrm{cov}[Z_0, Z_2] & \dots & \mathrm{cov}[Z_0, Z_{n_s}] \end{bmatrix}$$

Toutes les autres grandeurs ne sont à calculer qu'une seule fois lors de la construction du modèle. C'est pour cette raison que, malgré un coût de construction pouvant être important (mais qui reste négligeable face au temps de calcul du problème mécanique), l'évaluation d'un métamodèle de krigeage est extrêmement peu coûteuse (elle se réduit finalement à des produits scalaires).

La détermination de la variance de krigeage permet de construire des intervalles de confiance. Définis tels que $\forall \mathbf{x}^{(0)},\ I_c^\Delta = [\tilde{y}_\bullet\big(\mathbf{x}^{(0)}\big) - \Delta s_\bullet\big(\mathbf{x}^{(0)}\big), \tilde{y}_\bullet\big(\mathbf{x}^{(0)}\big) + \Delta s_\bullet\big(\mathbf{x}^{(0)}\big)]$, ils permettent de dire que la fonction prend en $\mathbf{x}^{(0)}$ une valeur dans l'intervalle I_c avec une probabilité d'environ 68% pour $\Delta = 1$, d'environ 95% pour $\Delta = 2$ et d'environ 99.7% pour $\Delta = 3$.

Remarque 3.9 : Dans le cadre du krigeage universel, les coefficients apparaissant dans le modèle de tendance et regroupés dans le vecteur $\hat{\boldsymbol{\beta}}$ peuvent être vus comme l'estimation de $\boldsymbol{\beta}$ par la méthode des moindres carrés généralisés.

2.1.4 Structure de covariance

Comme on l'a vu dans les formulations précédentes, il est nécessaire, pour parvenir à élaborer un modèle de krigeage, d'être capable de construire la matrice $\mathbf{C}$ et le vecteur $\mathbf{c}_0$. Ces derniers font respectivement intervenir les covariances entre les réponses aux sites

échantillonnés et entre les réponses aux sites échantillonnés et au point courant $\mathbf{x}^{(0)}$. Dans la pratique, il est extrêmement rare de connaître *a priori* la véritable forme de la structure de covariance. C'est pourquoi on utilise classiquement une fonction de covariance k (ou fonction noyau ou noyau de covariance) pour représenter cette structure :

$$\begin{aligned}&\forall (i,j) \in [\![0, n_s]\!],\\ &\forall (\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \in \mathcal{D}^2,\ \mathrm{cov}\big[Z\big(\mathbf{x}^{(i)}\big), Z\big(\mathbf{x}^{(j)}\big)\big] = k\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\big) = \sigma_Z^2 R\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \mathbf{l}\big)\end{aligned} \quad (3.38)$$

où R est une fonction de corrélation paramétrique de paramètres $\mathbf{l}$ ($\mathbf{l} = \begin{bmatrix}\ell_1 & \dots & \ell_{n_p}\end{bmatrix}^T$). Ces paramètres sont classiquement désignés sous le terme « longueurs de corrélation ». Cette fonction est symétrique, définie positive, vérifie $R(\mathbf{x}, \mathbf{x}; \mathbf{l}) = 1$ et est de la forme

$$R\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \mathbf{l}\big) = \prod_{k=1}^{n_p} h(x_k^{(i)} - x_k^{(j)}; \ell_k) \quad (3.39)$$

Sous cette formulation, la fonction de corrélation garde les propriétés des fonctions de corrélation unidimensionnelles $h(r)$. Le tableau 3.2 présente les fonctions de corrélation classiquement employées [C. E. RASMUSSEN et C. K. I. WILLIAMS 2006].

Noms	$h(r)$	Paramètres
Linéaire	$\max\left(0, 1 - \dfrac{\lvert r\rvert}{\ell}\right)$	$\ell > 0$
Gaussienne	$\exp\left(-\dfrac{r^2}{2\ell^2}\right)$	$\ell > 0$
Exponentielle généralisée	$\exp\left(-\left(\dfrac{\lvert r\rvert}{\ell}\right)^p\right)$	$\ell > 0\ 0 \leq p \leq 2$
Spline cubique	$\begin{cases} 1 - 6\left(\dfrac{r}{\ell}\right)^2 + \left(\dfrac{\lvert r\rvert}{\ell}\right)^3 & \text{si } \lvert r\rvert < \dfrac{\ell}{2} \\ 2\left(1 - \dfrac{\lvert r\rvert}{\ell}\right) & \text{si } \dfrac{\ell}{2} \leq \lvert r\rvert < \ell \\ 0 & \text{si } \lvert r\rvert \geq \ell \end{cases}$	$\ell > 0$
Matérn	$\dfrac{2^{1-\nu}}{\Gamma(\nu)}\left(\dfrac{\sqrt{2\nu}\lvert r\rvert}{\ell}\right)^{\nu} K_\nu\left(\dfrac{\sqrt{2\nu}\lvert r\rvert}{\ell}\right)$	$\ell > 0\ \nu > 0$

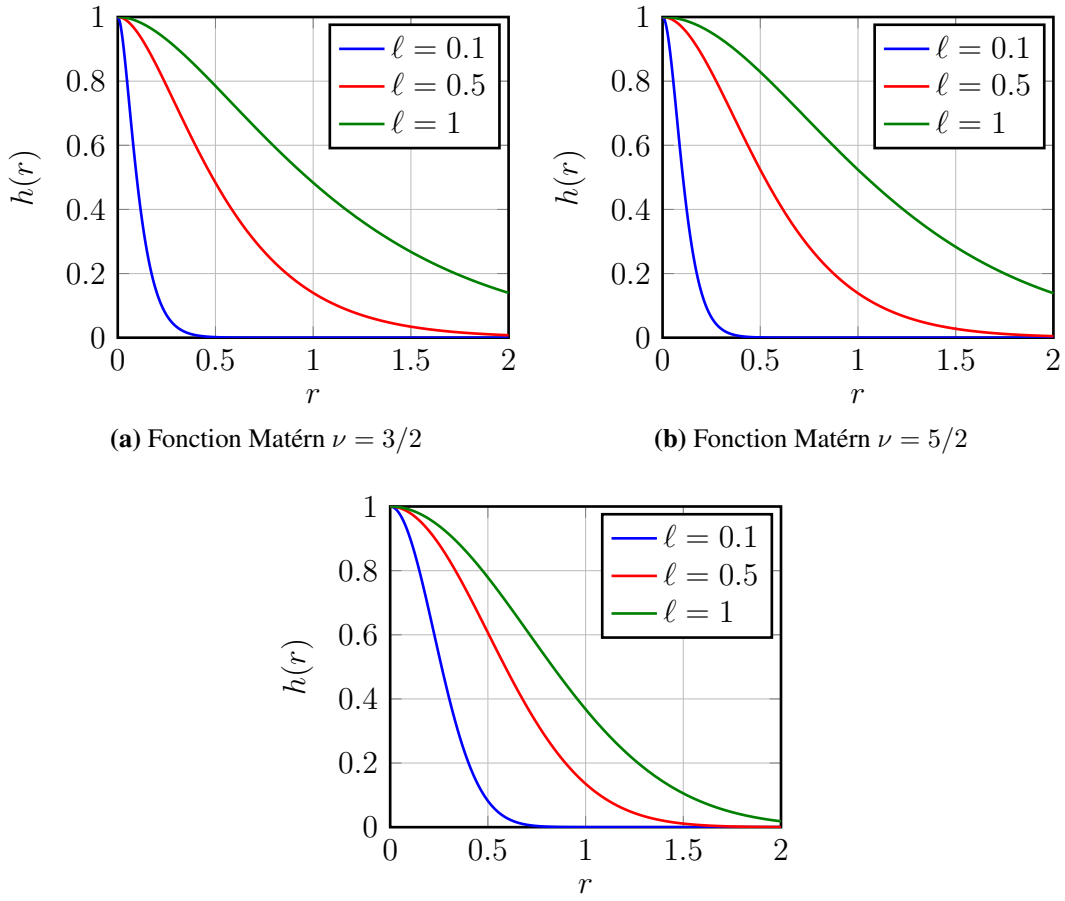
TABLE 3.2 – Quelques fonctions de corrélation

Généralement, et tel que cela a été introduit dans la méthode DACE [SACKS, WELCH et al. 1989], la fonction employée est la fonction gaussienne. Plus récemment, plusieurs travaux [STEIN 1999 ; C. E. RASMUSSEN et C. K. I. WILLIAMS 2006] ouvrent la voie à l'usage de la fonction Matérn [MATÉRN 1960]. Cette dernière, construite à partir de la fonction gamma Γ et de la fonction de Bessel modifiée K_ν présente la particularité de former en réalité un ensemble de fonctions dont le comportement est contrôlé par les

paramètres ℓ et ν. En particulier, le choix de la valeur de ν permet d'ajuster la régularité du processus gaussien associé (ce processus est $\lceil \nu - 1 \rceil$ différentiable [2]). Dans le cadre de ces travaux, on ne s'intéressera qu'à l'usage des formes simplifiées de cette fonction prises pour $\nu = \frac{3}{2}$ et $\nu = \frac{5}{2}$.

$$h_{\nu=3/2}(r) = \left(1 + \frac{\sqrt{3}|r|}{\ell}\right) \exp\left(-\frac{\sqrt{3}|r|}{\ell}\right) \tag{3.40}$$

$$h_{\nu=5/2}(r) = \left(1 + \frac{\sqrt{5}|r|}{\ell} + \frac{5r^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}|r|}{\ell}\right) \tag{3.41}$$



(a) Fonction Matérn $\nu = 3/2$

(b) Fonction Matérn $\nu = 5/2$

(c) Fonction Gaussienne

FIGURE 3.2 – Fonctions de corrélation Gaussienne et Matérn pour 3 valeurs de la portée ℓ

Finalement, la matrice de covariance $\mathbf{C}$, symétrique, définie, positive, peut être construite à partir de ces fonctions de corrélation et être exprimée en fonction d'une

2. La fonction $\lceil \bullet \rceil$ est définie telle que $\forall x \in \mathbb{R},\ \lceil x \rceil = \max(m \in \mathbb{Z} | m \leq x)$

matrice de corrélation $\mathbf{K}$.

$$\begin{aligned}\mathbf{C} &= \begin{bmatrix} k\left(\mathbf{x}^{(1)},\mathbf{x}^{(1)}\right) & k\left(\mathbf{x}^{(1)},\mathbf{x}^{(2)}\right) & \dots & k\left(\mathbf{x}^{(1)},\mathbf{x}^{(n_s)}\right) \\ k\left(\mathbf{x}^{(2)},\mathbf{x}^{(1)}\right) & k\left(\mathbf{x}^{(2)},\mathbf{x}^{(2)}\right) & & \vdots \\ \vdots & & \ddots & \\ k\left(\mathbf{x}^{(n_s)},\mathbf{x}^{(1)}\right) & k\left(\mathbf{x}^{(n_s)},\mathbf{x}^{(2)}\right) & \dots & k\left(\mathbf{x}^{(n_s)},\mathbf{x}^{(n_s)}\right) \end{bmatrix} \\ &= \sigma_Z^2 \begin{bmatrix} R\left(\mathbf{x}^{(1)},\mathbf{x}^{(1)}\right) & R\left(\mathbf{x}^{(1)},\mathbf{x}^{(2)}\right) & \dots & R\left(\mathbf{x}^{(1)},\mathbf{x}^{(n_s)}\right) \\ R\left(\mathbf{x}^{(2)},\mathbf{x}^{(1)}\right) & R\left(\mathbf{x}^{(2)},\mathbf{x}^{(2)}\right) & & \vdots \\ \vdots & & \ddots & \\ R\left(\mathbf{x}^{(n_s)},\mathbf{x}^{(1)}\right) & R\left(\mathbf{x}^{(n_s)},\mathbf{x}^{(2)}\right) & \dots & R\left(\mathbf{x}^{(n_s)},\mathbf{x}^{(n_s)}\right) \end{bmatrix} = \sigma_Z^2\mathbf{K}\end{aligned} \tag{3.42}$$

De la même manière, il est possible de formuler le vecteur de covariance $\mathbf{c}_0$ en fonction du vecteur de corrélation $\mathbf{r}_0$.

$$\begin{aligned}\mathbf{c}_0 &= \begin{bmatrix} k\left(\mathbf{x}^{(0)},\mathbf{x}^{(1)}\right) & k\left(\mathbf{x}^{(0)},\mathbf{x}^{(1)}\right) & \dots & k\left(\mathbf{x}^{(0)},\mathbf{x}^{(n_s)}\right) \end{bmatrix}^T \\ &= \sigma_Z^2 \begin{bmatrix} R\left(\mathbf{x}^{(0)},\mathbf{x}^{(1)}\right) & R\left(\mathbf{x}^{(0)},\mathbf{x}^{(1)}\right) & \dots & R\left(\mathbf{x}^{(0)},\mathbf{x}^{(n_s)}\right) \end{bmatrix}^T = \sigma_Z^2\mathbf{r}_0\end{aligned} \tag{3.43}$$

Ces nouvelles notations permettent de reformuler les estimateurs et variances de krigeage sous les formes données par le tableau 3.3.

	Type	Formulation	Avec
$\tilde{y}_\bullet\left(\mathbf{x}^{(0)}\right)$	KS	$m + \mathbf{r}_0^T\mathbf{K}^{-1}\left(\mathbf{y}_{n_s} - m\mathbf{1}_{n_s}\right)$	–
	KO	$\hat{\beta} + \mathbf{r}_0^T\mathbf{K}^{-1}\left(\mathbf{y}_{n_s} - \hat{\beta}\mathbf{1}_{n_s}\right)$	$\hat{\beta} = \left(\mathbf{1}_{n_s}^T\mathbf{K}^{-1}\mathbf{1}_{n_s}\right)^{-1}\mathbf{1}_{n_s}^T\mathbf{K}^{-1}\mathbf{y}_{n_s}$
	KU	$\mathbf{f}_0^T\hat{\boldsymbol{\beta}} + \mathbf{r}_0^T\mathbf{K}^{-1}\left(\mathbf{y}_{n_s} - \mathbf{F}\hat{\boldsymbol{\beta}}\right)$	$\hat{\boldsymbol{\beta}} = \left(\mathbf{F}^T\mathbf{K}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{K}^{-1}\mathbf{y}_{n_s}$
$s_\bullet^2\left(\mathbf{x}^{(0)}\right)$	KS	$\sigma_Y^2\left(1 - \mathbf{r}_0^T\mathbf{K}^{-1}\mathbf{r}_0\right)$	–
	KO	$\sigma_Z^2\left(1 - \mathbf{r}_0^T\mathbf{K}^{-1}\mathbf{r}_0 + \mathbf{u}_0^T\left(\mathbf{1}_{n_s}^T\mathbf{K}^{-1}\mathbf{1}_{n_s}\right)^{-1}\mathbf{u}_0\right)$	$\mathbf{u}_0 = \mathbf{1}_{n_s}^T\mathbf{K}^{-1}\mathbf{r}_0 - 1$
	KU	$\sigma_Z^2\left(1 - \mathbf{r}_0^T\mathbf{K}^{-1}\mathbf{r}_0 + \mathbf{u}_0^T\left(\mathbf{F}^T\mathbf{K}^{-1}\mathbf{F}\right)^{-1}\mathbf{u}_0\right)$	$\mathbf{u}_0 = \mathbf{F}^T\mathbf{K}^{-1}\mathbf{r}_0 - \mathbf{f}_0$

TABLE 3.3 – Approximations et variances du krigeage exprimées à l'aide des fonctions de corrélation

2.1.5 Interpolation

Afin de vérifier le comportement de l'approximation fournie par le krigeage aux points échantillonnés $\mathbf{x}^{(i)}$ ($i \in [\![1, n_s]\!]$), calculons les valeurs de l'approximation et de la variance en ces points. Ainsi, dans le cas du krigeage universel,

$$\begin{aligned}\tilde{y}_{KU}\left(\mathbf{x}^{(i)}\right) &= \mathbf{f}_i^T\hat{\boldsymbol{\beta}} + \mathbf{r}_i^T\mathbf{K}^{-1}\left(\mathbf{y}_{n_s} - \mathbf{F}\hat{\boldsymbol{\beta}}\right) \\ s_{KU}^2\left(\mathbf{x}^{(i)}\right) &= \sigma_Z^2\left(1 - \mathbf{r}_i^T\mathbf{K}^{-1}\mathbf{r}_i + \mathbf{u}_i^T\left(\mathbf{F}^T\mathbf{K}^{-1}\mathbf{F}\right)^{-1}\mathbf{u}_i\right) \text{ avec } \mathbf{u}_i = \mathbf{F}^T\mathbf{K}^{-1}\mathbf{r}_i - \mathbf{f}_i\end{aligned}$$

Le vecteur $\mathbf{r}_i$ correspond à la $i-$ème colonne de la matrice $\mathbf{K}$ tandis que le vecteur $\mathbf{f}_i$ correspond à la $i-$ème colonne de la matrice $\mathbf{F}^T$. Ainsi, on obtient $\mathbf{K}^{-1}\mathbf{r}_i = \mathbf{e}_i$ où $\mathbf{e}_i$ est le $i-$ème vecteur de base de $\mathbb{R}^{n_s}$. Il est alors constitué de 0 à l'exception de la $i-$ème composante qui vaut 1.

Ainsi,

$$\mathbf{r}_i^T\mathbf{K}^{-1}\left(\mathbf{y}_{n_s} - \mathbf{F}\hat{\boldsymbol{\beta}}\right) = \mathbf{e}_i^T\left(\mathbf{y}_{n_s} - \mathbf{F}\hat{\boldsymbol{\beta}}\right) = y_i - \mathbf{f}_i^T\hat{\boldsymbol{\beta}}$$

$$\mathbf{u}_i = \mathbf{F}^T\mathbf{K}^{-1}\mathbf{r}_i - \mathbf{f}_i = \mathbf{f}_i - \mathbf{f}_i = \mathbf{0}_p \qquad \mathbf{r}_i^T\mathbf{K}^{-1}\mathbf{r}_i = \mathbf{e}_i^T\mathbf{r}_i = R\big(\mathbf{x}^{(i)}, \mathbf{x}^{(i)}\big) = 1$$

Finalement, on obtient

$$\begin{aligned}
&\forall i \in [\![1, n_s]\!], \forall \mathbf{x}^{(i)} \in \mathcal{D},\\
&\quad \tilde{y}_{KU}\big(\mathbf{x}^{(i)}\big) = y_i = y\big(\mathbf{x}^{(i)}\big)\\
&\quad s^2_{KU}\big(\mathbf{x}^{(i)}\big) = \sigma_Z^2\,(1-1) = 0
\end{aligned}$$

Le modèle de krigeage basé sur des réponses déterministes aux points échantillonnés fourni alors une approximation interpolante. Par ailleurs, la variance est nulle aux points échantillonnés. Ces résultats sont également vrais pour le krigeage simple, le krigeage ordinaire et le krigeage universel.

2.1.6 Illustration du krigeage

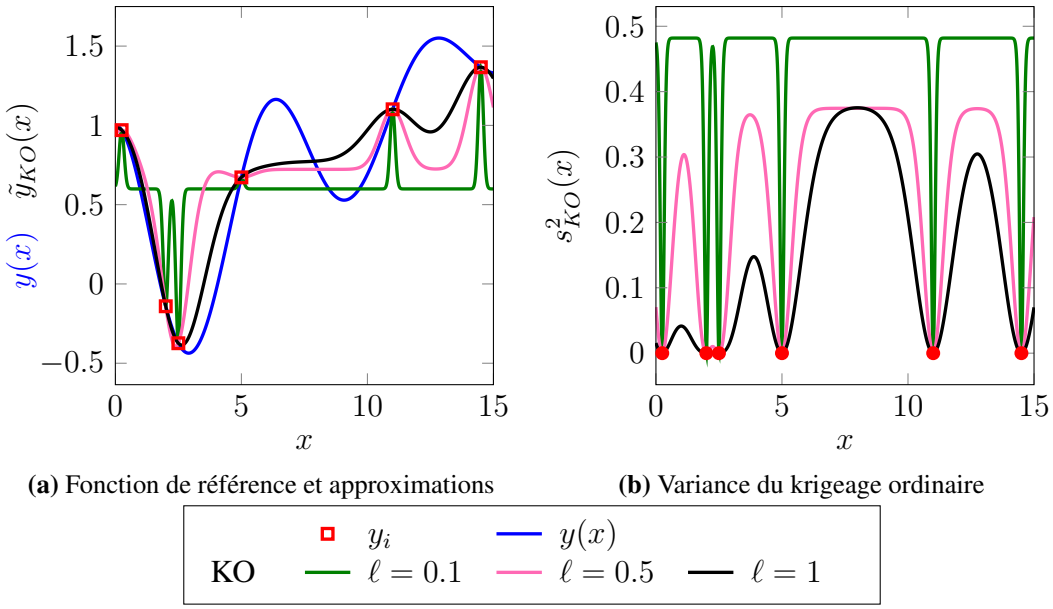
Afin d'illustrer le comportement du krigeage, considérons une fonction analytique à une variable $y(x) = \exp(-x/10)\cos(x) + x/10$. La figure 3.3 présente l'approximation et la variance fournies par un métamodèle de krigeage ordinaire. Les deux figures sont construites en considérant une fonction de corrélation de type gaussienne et 3 valeurs du paramètre ℓ.

On constate que le paramètre du modèle de krigeage a un impact considérable sur la qualité de la prédiction. C'est pourquoi, par la suite, on emploiera une stratégie dédiée à la détermination de ces paramètres.

Par ailleurs, en fixant un des paramètres, il est possible de faire évoluer le nombre de réponses y_i utilisées pour construire l'approximation. La figure 3.4 présente les approximations et variances fournies par des métamodèles de krigeage ordinaire pour une longueur de corrélation fixée à 1 et pour 3 tirages. On constate, comme on pouvait l'attendre, que l'augmentation du nombre de points échantillonnés permet d'obtenir une approximation de meilleure qualité. Par ailleurs, la variance du krigeage se trouve diminuée également.

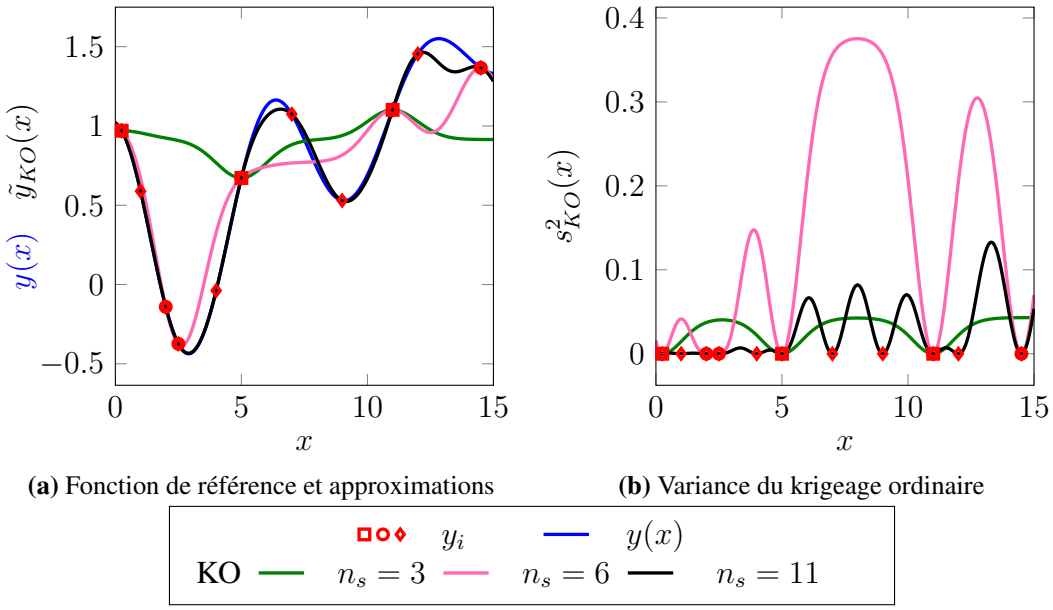
2.1.7 Détermination des paramètres

La formulation du krigeage fait apparaître un certain nombre de paramètres (que l'on désigne parfois sous le terme d'*hyperparamètres*) tels que σ_Z^2 et $\mathbf{l}$ qu'il est nécessaire de fixer. En effet, ceux-ci on un impact non négligeable (voir la figure 3.3) sur la qualité d'approximation de la méthode. Il existe plusieurs méthodes qui sont brièvement décrites dans la suite.



(a) Fonction de référence et approximations **(b)** Variance du krigeage ordinaire

FIGURE 3.3 – Approximations et variances obtenues par krigeage ordinaire pour 3 paramètres de corrélation (fonction de corrélation gaussienne)



(a) Fonction de référence et approximations **(b)** Variance du krigeage ordinaire

FIGURE 3.4 – Approximations et variances obtenues par krigeage ordinaire pour 3 tirages comportant 3, 6 et 10 points (fonction de corrélation gaussienne, paramètre $\ell = 1$)

- Dans un contexte géostatistique, la méthode consiste à employer un **semi-variogramme** permettant de lier la variance de la différence des réponses $y\left(\mathbf{x}^{(i)}\right)$ et $y\left(\mathbf{x}^{(j)}\right)$ à la distance entre les sites associés $\mathbf{x}^{(i)}$ et $\mathbf{x}^{(j)}$. Généralement non connu *a priori*, le semi-variogramme réel est remplacé par un semi-variogramme empirique construit à partir des réponses obtenues aux points échantillonnés. À ces derniers, la méthode fait correspondre un semi-variogramme de référence basé sur l'emploi d'une fonction paramétrique dont les paramètres sont obtenus en employant une méthode des moindres carrés (ordinaires, pondérés ou généralisés). On pourra se référer à [CRESSIE 1993] pour des détails concernant la mise en œuvre de cette méthode.

  Remarque 3.10 : L'usage des semi-variogrammes permet par ailleurs l'obtention d'informations complémentaires comme la présence de bruit d'observation (on obtient alors sa variance), la régularité de la sortie du système, la portée (au-delà de laquelle le semi-variogramme devient approximativement constant) et la variance des observations.

- la méthode actuellement la plus employée s'appuie sur le cadre statistique du krigeage et en particulier sur celui des processus gaussiens. Elle propose de déterminer les paramètres du modèle en maximisant une fonction nommée vraisemblance. Cette technique connue sous le nom de **Maximum de vraisemblance** a été utilisée pour déterminer la covariance d'un processus aléatoire spatial [KITANIDIS 1983 ; MARDIA et MARSHALL 1984 ; R. H. JONES et VECCHIA 1993]. L'idée repose sur la recherche des paramètres du modèle de sorte que l'observation des réponses soit la plus vraisemblable possible, c'est-à-dire tel que l'on maximise la probabilité des réponses. Dans le cadre d'un krigeage universel introduit dans la partie 2.1.1 page 135 et en écrivant la matrice de corrélation sous la forme $\mathbf{K} = \mathbf{K}(\mathbf{l})$, la vraisemblance s'écrit (pour les autres formes de krigeage, on pourra aisément modifier la formule en utilisant par exemple le tableau 3.3) :

$$L(\boldsymbol{\beta}, \sigma_Z^2, \mathbf{l}) = \left(2\pi\sigma_Z^2\right)^{-n_s/2} |\mathbf{K}(\mathbf{l})|^{-1/2} \exp\left[-\frac{1}{2\sigma_Z^2}(\mathbf{Y}_{n_s} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{K}(\mathbf{l})^{-1}(\mathbf{Y}_{n_s} - \mathbf{F}\boldsymbol{\beta})\right] \quad (3.44)$$

Dans la pratique, on emploie une formulation de log-vraisemblance. La détermination se fera alors par minimisation de la fonction :

$$-\log L(\boldsymbol{\beta}, \sigma_Z^2, \mathbf{l}) = \frac{n_s}{2}\log(\pi) + \frac{n_s}{2}\log(\sigma_Z^2) + \frac{1}{2}\log(|\mathbf{K}(\mathbf{l})|) \quad (3.45)$$

$$+ \frac{1}{2\sigma_Z^2}(\mathbf{Y}_{n_s} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{K}(\mathbf{l})^{-1}(\mathbf{Y}_{n_s} - \mathbf{F}\boldsymbol{\beta}) \quad (3.46)$$

Ainsi (où $\mathbb{L}$ désigne l'espace des paramètres du krigeage)

$$(\hat{\boldsymbol{\beta}}, \hat{\sigma}_Z^2, \hat{\mathbf{l}}) = \underset{(\boldsymbol{\beta}, \sigma_Z^2, \mathbf{l}) \in \mathbb{R}^p \times \mathbb{R}^{*+} \times \mathbb{L}}{\arg\min} -\log L(\boldsymbol{\beta}, \sigma_Z^2, \mathbf{l}) \quad (3.47)$$

La fonction $L(\boldsymbol{\beta}, \sigma_Z^2, \mathbf{l})$ étant définie explicitement en fonction de $\boldsymbol{\beta}$ et σ_Z^2, il est possible de dériver la vraisemblance par rapport à ces variables et d'en déduire les valeurs $\hat{\boldsymbol{\beta}}$ et $\hat{\sigma}_Z^2$ de $\boldsymbol{\beta}$ et σ_Z^2 qui maximisent la vraisemblance

$$\left|\begin{array}{l} \dfrac{\partial L}{\partial \boldsymbol{\beta}}(\boldsymbol{\beta}, \sigma_Z^2, \mathbf{l}) = \mathbf{0} \Longrightarrow \forall (\sigma_Z^2, \mathbf{l}) \in \mathbb{R}^{*+} \times \mathbb{L}, \\ \qquad\qquad \dfrac{1}{\sigma_Z^2}\left(\mathbf{F}^T \mathbf{K}(\mathbf{l})^{-1} \mathbf{Y}_{n_s} - \mathbf{F}^T \mathbf{K}(\mathbf{l})^{-1} \mathbf{F} \boldsymbol{\beta}\right) = \mathbf{0} \\ \dfrac{\partial L}{\partial \sigma_Z^2}(\boldsymbol{\beta}, \sigma_Z^2, \mathbf{l}) = \mathbf{0} \Longrightarrow \forall (\boldsymbol{\beta}, \mathbf{l}) \in \mathbb{R}^{*+} \times \mathbb{L}, \\ \qquad\qquad -\dfrac{n_s}{2\sigma_Z^2} + \dfrac{1}{2\sigma_Z^4}(\mathbf{Y}_{n_s} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{K}(\mathbf{l})^{-1}(\mathbf{Y}_{n_s} - \mathbf{F}\boldsymbol{\beta}) = 0 \end{array}\right. \tag{3.48}$$

Finalement, on obtient

$$\hat{\boldsymbol{\beta}}(\mathbf{l}) = \left(\mathbf{F}^T \mathbf{K}(\mathbf{l})^{-1} \mathbf{F}\right)^{-1} \mathbf{F}^T \mathbf{K}(\mathbf{l})^{-1} \mathbf{Y}_{n_s} \tag{3.49}$$

$$\hat{\sigma}_Z^2(\mathbf{l}) = \frac{1}{n_s}(\mathbf{Y}_{n_s} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{K}(\mathbf{l})^{-1}(\mathbf{Y}_{n_s} - \mathbf{F}\boldsymbol{\beta}) \tag{3.50}$$

On retrouve bien le vecteur des coefficients $\hat{\boldsymbol{\beta}}$ vu précédemment (cf. tableau 3.3 page 146). Les paramètres $\mathbf{l}$ du krigeage apparaissant de manière implicite lors du calcul de la matrice de corrélation $\mathbf{K}(\mathbf{l})$, l'usage d'une méthode d'optimisation numérique est alors nécessaire. En injectant $\hat{\boldsymbol{\beta}}$ et $\hat{\sigma}_Z^2$ dans l'expression de la vraisemblance, les paramètres sont cherchés tels que

$$\hat{\mathbf{l}} = \underset{\mathbf{l} \in \mathbb{L}}{\arg\min}\ \psi(\mathbf{l}) \text{ avec } \psi(\mathbf{l}) = \hat{\sigma}_Z^2(\mathbf{l}) |\mathbf{K}(\mathbf{l})|^{1/n_s} \tag{3.51}$$

La recherche de ce minimum requiert l'usage d'un algorithme adapté et peut s'avérer complexe en raison :

- ▷ de la présence possible d'optima locaux [WARNES et RIPLEY 1987] nécessitant soit d'employer une méthode d'optimisation globale, par exemple un algorithme génétique, comme recommandé dans [FORRESTER, SÓBESTER et KEANE 2008] soit un algorithme génétique utilisant les gradients comme pour le package DiceKriging [3] pour R. La toolbox DACE [LOPHAVEN, NIELSEN et SØNDERGAARD 2002] pour MATLAB utilise quant à elle un algorithme BOXMIN s'apparentant à une méthode de dichotomie multidimensionnelle. Au besoin, il est possible de formuler analytiquement les dérivées de la vraisemblance par rapport aux paramètres en fonction de la matrice $\mathbf{K}$ (cf. [VAZQUEZ 2005, p. 161]). Par ailleurs, de longues plages de valeurs égales peuvent rendre la convergence de l'algorithme relativement lente.
- ▷ de la nécessité, pour certains paramètres de la fonction de corrélation, de tenir compte des contraintes portant sur ceux-ci.

3. `http://cran.r-project.org/web/packages/DiceKriging/DiceKriging.pdf`

- ▷ de l'importance du coût nécessaire à l'évaluation de la vraisemblance. L'utilisation de cette méthode lorsque l'on considère un nombre élevé de points échantillonnés (par exemple pour $n_s > 1000$) nécessite l'usage de méthodes adaptées capables d'évaluer la vraisemblance en utilisant qu'une partie des échantillons [STEIN, CHI et WELTY 2004].

L'emploi d'une factorisation de la matrice de corrélation $\mathbf{K}$, par exemple de type Cholesky (possible ici en raison de la positivité de la matrice de corrélation), permet de faciliter à la fois les opérations liées au calcul du krigeage et le calcul du déterminant qui est obtenu en faisant le produit des éléments diagonaux de la matrice triangulaire obtenue lors de la factorisation.

- ▶ Dans un contexte plus général des modèles paramétriques (et non nécessairement basés sur un contexte probabiliste), les méthodes de validation croisée sont largement employées. Le principe déjà abordé dans la partie 4.6.6 du chapitre 1 page 87 consiste à supprimer successivement une ou plusieurs réponses(s) disponible(s) aux points échantillonnés, à construire le modèle de substitution en absence de cette ou ces information(s) puis à évaluer l'écart entre la ou les réponse(s) non prise(s) en compte et celle(s) estimée(s) par le modèle. En notant $\tilde{y}_{-i}$ l'approximation fournie par le modèle de substitution en absence de la réponse $y\big(\mathbf{x}^{(i)}\big)$, la validation croisée (dans ce cas désignée sous le nom *Leave One-Out*) se calcule de la manière suivante :

$$\mathrm{LOO}(\mathrm{l}) = \sum_{i=1}^{n_s} \big(\tilde{y}_{-i}(\mathbf{x}^{(i)}) - y\big(\mathbf{x}^{(i)}\big)\big)^2 \tag{3.52}$$

La méthode consiste à chercher les paramètres $\hat{\mathrm{l}}$ tels que

$$\hat{\mathrm{l}} = \arg\min_{\mathrm{l}\in\mathbb{L}} \mathrm{LOO}(\mathrm{l}) \tag{3.53}$$

Cette approche est particulièrement coûteuse en raison notamment de la nécessité de construire n_s modèles de substitution pour évaluer cette erreur. C'est pourquoi Rippa propose, dans le cadre d'un modèle de substitution de type RBF, une méthode efficace de calcul de ce critère [RIPPA 1999]. M. Bompard propose l'adaptation de cette méthode au krigeage [BOMPARD 2011, page 149 (annexe B.2)] et obtient ainsi :

$$\mathrm{LOO}(\mathrm{l}) = \frac{\mathbf{E}^T\mathbf{E}}{n_s} \text{ avec } \mathbf{E} = \begin{bmatrix} e_1 & e_2 & \dots & e_{n_s} \end{bmatrix}^T \text{ où } e_i = \left| \frac{(\mathbf{\Lambda}^{-1}\mathbf{y}_+)_i}{(\mathbf{\Lambda}^{-1})_{ii}} \right| \tag{3.54}$$

où dans le cas du krigeage universel, $\mathbf{\Lambda} = \begin{bmatrix} \mathbf{C} & \mathbf{F} \\ \mathbf{F}^T & 0 \end{bmatrix}$ et $\mathbf{y}_+ = \begin{bmatrix} \mathbf{y}_{n_s}^T & \mathbf{0}_p^T \end{bmatrix}^T$.

Dans le cadre de l'approximation de fonctions obtenues par l'intermédiaire d'un code de calcul, les méthodes du maximum de vraisemblance et de validation croisée sont les plus employées. Quelques travaux [WAHBA 1985 ; STEIN 1990 ; WAHBA 1990 ; D.

ZIMMERMAN et M. ZIMMERMAN 1991] cherchent à comparer ces méthodes. Cependant, le choix s'avère particulièrement ardu. En effet, les performances de ces méthodes sont directement impactées par le type de fonction de covariance, le nombre de paramètres (n_p) du problème et le nombre d'échantillons pris en compte (n_s). Les problèmes pris en compte dans ces travaux comportent entre quelques dizaines et quelques centaines d'échantillons. Pour des dimensions allant de 1 à une dizaine de paramètres, il a été choisi d'utiliser la méthode du maximum de vraisemblance pour déterminer les paramètres.

■ **Exemple à une variable :** On se propose de considérer la même fonction à une variable que celle présentée dans la partie 2.1.6 page 147. L'espace $[0, 15]$ où varie le paramètre sera échantillonné à l'aide d'une méthode LHS comportant 6 points. La figure 3.5 présente la réponse obtenue en cherchant la longueur de corrélation maximisant la vraisemblance pour un métamodèle de krigeage ordinaire.

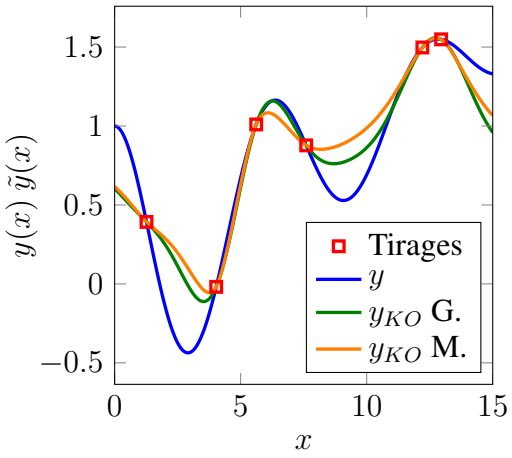
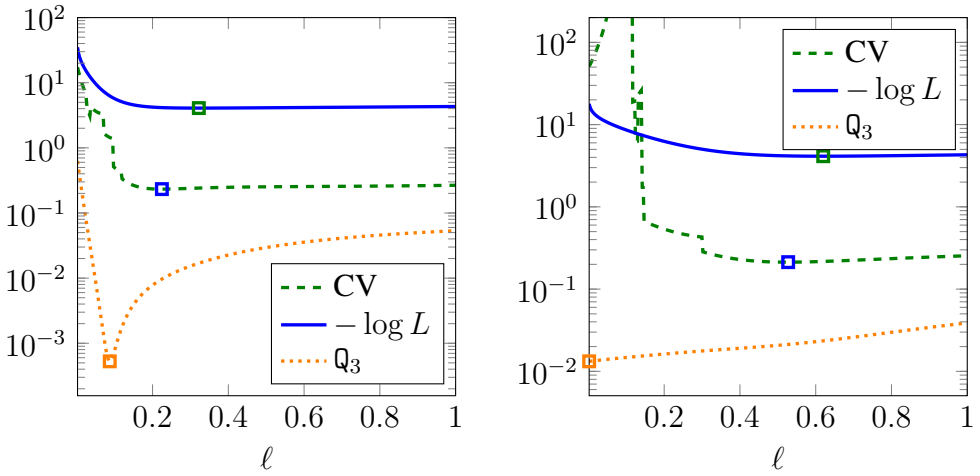


FIGURE 3.5 – Approximations par krigeage ordinaire (G. : corrélation gaussienne, M. : corrélation Matérn ($\nu = 3/2$))

Dans ce cas, où le nombre de sites échantillonnés permet d'obtenir une bonne représentativité de la fonction à approximer, le krigeage ordinaire permet d'obtenir une très bonne approximation.

La figure 3.6 présente l'évolution de la vraisemblance et de l'erreur de validation croisée en fonction du paramètre du krigeage. On observe que sur ce cas à une variable, la fonction Gaussienne fournit une approximation de meilleure qualité que la fonction Matérn. Ce constat ne s'est pas confirmé pour des dimensions supérieures pour lesquelles la fonction Matérn permettait d'obtenir une meilleure approximation. Globalement, la méthode du maximum de vraisemblance présente une meilleure robustesse par rapport au type de fonction de corrélation choisie. C'est pour cette raison qu'on utilisera cette méthode dans le reste de ces travaux.



(a) Fonction Gaussienne **(b)** Fonction Matérn ($\nu = 3/2$)

FIGURE 3.6 – Erreur d'approximation ($\mathbb{Q}_3$ voir page 87), erreur de validation croisée (CV) et -log-vraisemblance

2.1.8 Calculs des gradients du krigeage

Il est bien entendu possible de formuler les gradients de l'approximation par krigeage. La formulation n'est pas compliquée mais nécessite la dérivation des fonctions de corrélation. Il est donc nécessaire que celles-ci soient au moins $\mathcal{C}^1$.

2.1.9 Quelques aspects pratiques de la mise en œuvre du krigeage

La matrice de corrélation présente dans la construction du krigeage peut être le foyer de plusieurs problèmes. En effet, celle-ci doit être inversée lors de la résolution du problème de krigeage. Un mauvais conditionnement [4] peut alors rendre cette étape particulièrement complexe. Plusieurs causes et solutions à ce problème ont été identifiées. Parmi celles-ci on trouve :

- la présence de deux sites échantillonnés trop proches (dans l'espace des paramètres) pouvant engendrer l'apparition de deux lignes et colonnes de la matrice quasiment similaires. Le conditionnement s'en trouve alors dégradé et l'inversion peut devenir impossible. Pour remédier à ce problème, il est important d'employer une méthode de tirage assurant une bonne uniformité des points dans l'espace de conception et de ne pas avoir de sites en doublons.

4. Le conditionnement d'un matrice inversible $\mathbf{A}$ est défini par le nombre $\kappa(\mathbf{A}) = \|\mathbf{A}^{-1}\|\|\mathbf{A}\|$ pour une norme $\|\bullet\|$ (on a $\kappa(\mathbf{A}) \geq 1$). En considérant une norme euclidienne, on a également $\kappa(\mathbf{A}) = \mu_{\max}/\mu_1$ où les μ_i sont les valeurs singulières de $\mathbf{A}$, $\mu_1 = \min_i \mu_i$ et $\mu_{\max} = \max_i \mu_i$. On parle d'un bon conditionnement si $\kappa(\mathbf{A})$ est proche de 1 et d'un mauvais conditionnement si $\kappa(\mathbf{A})$ devient élevé (classiquement $\kappa(\mathbf{A}) > 10^{16}$ peut être considéré comme une limite au-delà de laquelle l'inversion de la matrice $\mathbf{A}$ n'est plus possible).

- certaines valeurs des paramètres peuvent altérer le conditionnement de la matrice de corrélation (voir figure 3.7b). Ce problème peut survenir lors de la détermination des paramètres du modèle de krigeage : par exemple lors de la maximisation de la vraisemblance par un algorithme d'optimisation, un jeu de paramètres de krigeage peut conduire à une matrice de corrélation mal conditionnée, à l'impossibilité de l'inversion et par conséquent à l'impossibilité du calcul de la vraisemblance. Il est donc nécessaire de définir judicieusement les bornes de l'espace où l'on recherche ces paramètres ou d'employer un algorithme d'optimisation ou une méthode capable de tenir compte de ce problème.

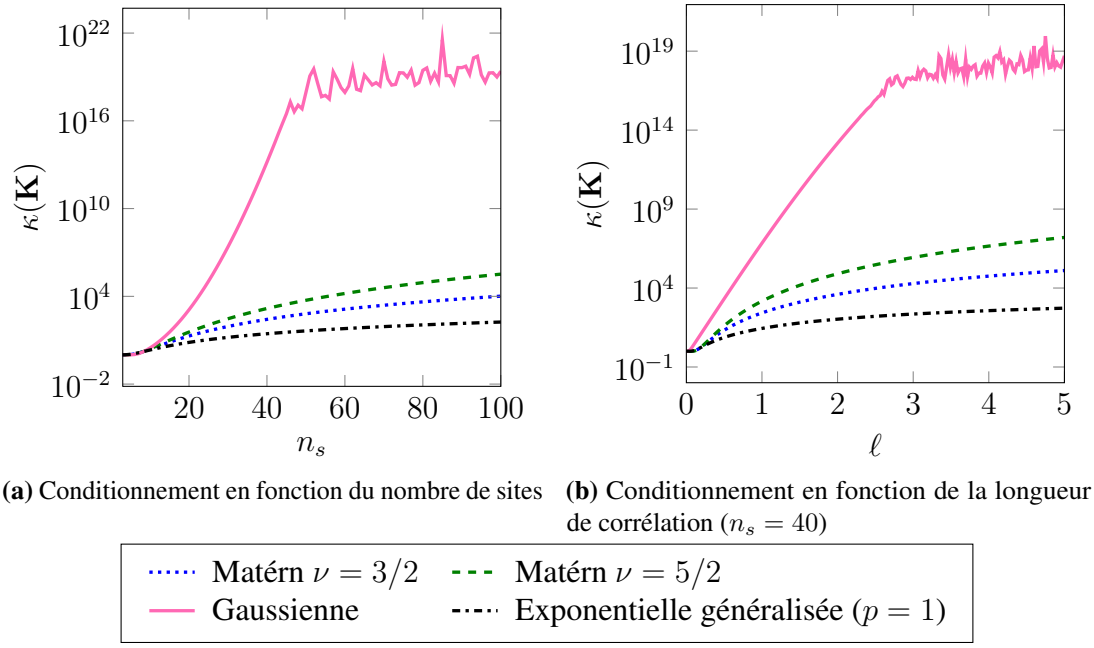
- parallèlement au problème précédent, certaines fonctions de corrélation peuvent engendrer une altération du conditionnement de la matrice de corrélation de manière plus systématique. Ce problème a été constaté par exemple dans [ABABOU, BAGTZOGLOU et WOOD 1994] où il est montré qu'en général, le conditionnement de la matrice de corrélation est mauvais pour une fonction gaussienne et meilleur pour une fonction exponentielle (voir la figure 3.7). La fonction Matérn permettrait quant à elle d'améliorer de façon notoire le conditionnement de la matrice de corrélation [VAZQUEZ 2005].

Ces différents problèmes relatifs au conditionnement de la matrice de corrélation vont croissants avec l'augmentation de la dimension du problème (lorsque le nombre n_s de sites échantillonnés augmente, voir la figure 3.7a). Une solution classiquement employée dans le cadre du krigeage consiste à ajouter à la matrice de corrélation une matrice diagonale du type $\delta \mathbf{I}_d$ où $\mathbf{I}_d$ est la matrice identité et $\delta \approx 10^{-15}$. Cette technique est utilisée dans la toolbox DACE [LOPHAVEN, NIELSEN et SØNDERGAARD 2002]. Le coefficient δ peut être considéré comme un bruit blanc. Il ne doit pas être choisi trop grand car il perturbe le caractère interpolant du krigeage qui ne vérifie alors plus la valeur de la réponse de la fonction aux sites échantillonnés. On pourra se tourner vers [RANJAN, HAYNES et KARSTEN 2011] pour des éléments de choix du paramètre δ et une version du krigeage corrigée pour assurer le caractère interpolant de la méthode.

La figure 3.7 montre que la fonction gaussienne peut présenter des problèmes de conditionnement conséquents. C'est pourquoi, dans la suite de ces travaux, le choix de la fonction s'est porté sur la fonction Matérn avec $\nu = 3/2$. En effet, bien que la fonction exponentielle généralisée permette d'obtenir de bonnes propriétés vis-à-vis du conditionnement de la matrice de corrélation, cette fonction comporte 2 paramètres complexifiant d'autant plus la phase de détermination des paramètres du krigeage.

Remarque 3.11 : D'une manière plus générale, l'ajout de termes sur la diagonale de la matrice de corrélation ou covariance se rapproche de la méthode mise en œuvre pour la prise en compte du bruit sur les observations [FORRESTER, KEANE et BRESSLOFF 2006 ; GINSBOURGER 2009]. On parle classiquement dans le cadre de la géostatistique d'effet pépite (ou *nugget effect*) pour désigner ce phénomène.

La maximisation de la vraisemblance (ou plutôt la minimisation de la -log-vraisemblance) sera réalisée à l'aide d'un algorithme PSO (*Particle Swarm Optimisation* cf. page 29). En



(a) Conditionnement en fonction du nombre de sites

(b) Conditionnement en fonction de la longueur de corrélation ($n_s = 40$)

FIGURE 3.7 – Conditionnement de la matrice de corrélation pour 4 fonctions de corrélation (échantillonnages réguliers de l'espace $[0, 15]$ et sans normalisation des sites)

effet, bien que le problème de minima locaux de la vraisemblance n'ait pas été constaté que sur des dimensions faibles (1, 2 et 3), des problèmes dans la détermination des paramètres du krigeage pour des dimensions supérieures dus à la présence de minima locaux ont été rencontrés. Le PSO permet de s'affranchir de ce problème. De plus, afin de faciliter cette phase de détermination des paramètres, une légère transformation des paramètres des fonctions de corrélation a été réalisée. Celle-ci consiste à ne plus considérer une fonction gaussienne du type $\exp(-r^2/(2\ell^2))$ mais une fonction de la forme $\exp(-\ell'^2 r^2/4)$. Ce changement assure une meilleure localisation du minimum de la vraisemblance et évite d'obtenir des vallées de paramètres tendant à minimiser la -log-vraisemblance. On procède à une modification du même type pour la fonction Matérn.

Finalement, pour l'ensemble des modèles de substitution construits (pour les modèles présentés par la suite également), une normalisation des données prises en compte par le modèle a été employée. Ainsi, tirages, réponses et gradients sont normalisés de manière à devenir des valeurs centrées réduites. Le métamodèle est alors construit à partir des points

échantillonnés $\mathbf{x}^{n(i)}$ et des réponses y_i^n construits de la manière suivante :

$$\forall (i,j) \in [\![1, n_s]\!] \times [\![1, n_p]\!], \tag{3.55}$$

$$x_j^{n(i)} = \frac{x_j^{(i)} - \overline{x_j}}{\sigma_{x_i}} \text{ où } \overline{x_j} = \frac{1}{n_s}\sum_{i=1}^{n_s} x_j^{(i)} \text{ et } \sigma_{x_i} = \sqrt{\frac{1}{n_s}\sum_{i=1}^{n_s}\left(x_j^{(i)} - \overline{x_j}\right)^2} \tag{3.56}$$

$$y_i^n = \frac{y_i - \overline{y}}{\sigma_y} \text{ où } \overline{y} = \frac{1}{n_s}\sum_{i=1}^{n_s} y_i \text{ et } \sigma_y = \sqrt{\frac{1}{n_s}\sum_{i=1}^{n_s}(y_i - \overline{y})^2} \tag{3.57}$$

Cette transformation permet de ne manipuler que des grandeurs de même ordre de grandeur de moyenne nulle et d'écart-type 1, ce qui améliore ainsi les différentes manipulations numériques de ces données lors de la construction du modèle de substitution. Une fois le modèle construit, on procède à la démarche inverse afin d'obtenir des réponses correspondant à leur bon ordre de grandeur.

2.2 Fonctions de base radiales (RBF)

Le principe des modèles de substitution basés sur les fonctions de base radiales consiste à chercher l'approximation sous la forme d'une combinaison linéaire de fonctions radiales. Il s'agit de fonctions dépendantes de la distance entre un point courant et un point central. Cette méthode fut introduite pour l'interpolation de fonctions de plusieurs variables [HARDY 1971 ; DUCHON 1977 ; M. J. POWELL 1981] et utilisée dans la résolution de problèmes aux dérivées partielles sans maillage. Par la suite, les fonctions de base radiales furent employées pour construire des réseaux de neurones [BROOMHEAD et LOWE 1988 ; MOODY et DARKEN 1989]. Dans ce contexte, on parle classiquement de réseau de fonctions de base radiales (*Radial Basis Functions Network* - RBF). Cette méthode se présente comme une alternative intéressante aux réseaux de neurones en raison principalement de leur coût de construction réduit. On pourra se référer aux travaux [ORR 1996 ; SCHABACK 1997 ; ORR 1999] pour de nombreux compléments relatifs aux RBF.

2.2.1 Formulation du modèle

Classiquement, pour un ensemble de n_s points et réponses associés $\left\{\mathbf{x}^{(i)}, y\left(\mathbf{x}^{(i)}\right)\right\}$, un modèle de substitution à base de RBF est cherché sous la forme :

$$\tilde{y}\left(\mathbf{x}^{(0)}\right) = \sum_{i=1}^{n_s} w_i \phi\left(\mathbf{x}^{(0)}, \mathbf{c}^{(i)}\right) = \mathbf{w}^T \boldsymbol{\phi}\left(\mathbf{x}^{(0)}\right) \tag{3.58}$$

où $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \dots & w_{n_s} \end{bmatrix}^T \in \mathbb{R}^{n_s}$ est le vecteur des coefficients (inconnues du problème), ϕ une fonction de base radiale et $\mathbf{c}^{(i)}$ un point de l'espace (appelé centre). Le choix des centres est un problème complexe pouvant être fait en employant par exemple une méthode de clustering [S. COHEN et INTRATOR 2000]. Pour des raisons de simplicité,

nous considérerons $\mathbf{c}^{(i)} = \mathbf{x}^{(i)}$. La construction du metamodèle RBF consiste à s'assurer que le modèle est interpolant soit :

$$\forall j \in [\![1, n_s]\!],\ \tilde{y}\big(\mathbf{x}^{(j)}\big) = \sum_{i=1}^{n_s} w_i \phi\big(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}\big) = y\big(\mathbf{x}^{(j)}\big) \tag{3.59}$$

Ainsi, ces n_s équations linéaires conduisent au système sous forme matricielle :

$$\mathbf{\Phi w} = \mathbf{y}_{n_s} \tag{3.60}$$

avec $\mathbf{\Phi} = \begin{bmatrix} \phi\big(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}\big) & \phi\big(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\big) & \dots & \phi\big(\mathbf{x}^{(1)}, \mathbf{x}^{(n_s)}\big) \\ \phi\big(\mathbf{x}^{(2)}, \mathbf{x}^{(1)}\big) & \phi\big(\mathbf{x}^{(2)}, \mathbf{x}^{(2)}\big) & & \vdots \\ \vdots & & \ddots & \\ \phi\big(\mathbf{x}^{(n_s)}, \mathbf{x}^{(1)}\big) & \dots & & \phi\big(\mathbf{x}^{(n_s)}, \mathbf{x}^{(n_s)}\big) \end{bmatrix}$.

Suite à cette résolution, l'approximation est alors obtenue de la manière suivante :

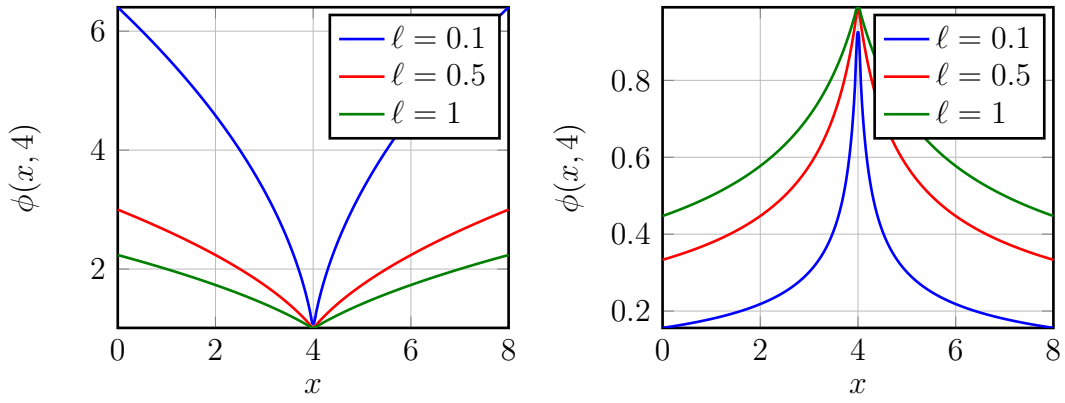
$$\boxed{\forall \mathbf{x}^{(0)} \in \mathcal{D},\ \tilde{y}\big(\mathbf{x}^{(0)}\big) = \mathbf{y}_{n_s}^T \mathbf{\Phi}^{-1} \boldsymbol{\phi}\big(\mathbf{x}^{(0)}\big)} \tag{3.61}$$

2.2.2 Choix de la fonction de base radiale

Parmi les fonctions de base radiales $\phi\big(\mathbf{x}, \mathbf{c}^{(i)}\big)$ classiquement employées, on trouve celles proposées dans le tableau 3.4. La figure 3.8 illustre le comportement de 3 fonctions radiales. On constate que ces fonctions peuvent avoir des comportements très différents (à la différences des fonctions de corrélation par exemple valant 1 en 0 et tendant vers 0 en s'éloignant de l'origine). D'autres fonctions peuvent être trouvées dans [SCHABACK 2007].

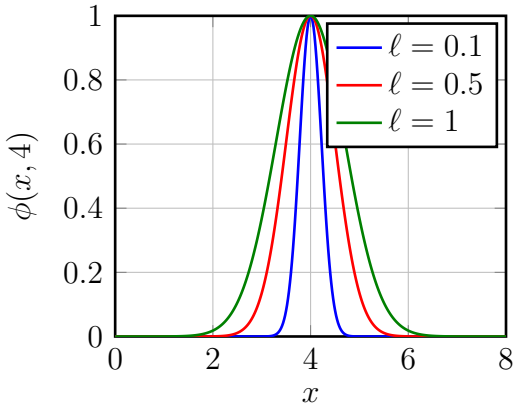
Noms	$\phi(\mathbf{x}, \mathbf{c})$	Paramètres
Splines linéaires	$\|\mathbf{x} - c_i\|$	–
Thin plate splines	$\|\mathbf{x} - \mathbf{c}^{(i)}\|^k \ln \|\mathbf{x} - \mathbf{c}^{(i)}\|$ $k = 2p + 1$ avec $p \in \mathbb{N}$	–
Splines cubiques	$\|\mathbf{x} - c_i\|^3$	–
Gaussienne	$\exp\left(-\dfrac{\|\mathbf{x} - \mathbf{c}^{(i)}\|^2}{\ell}\right)$	$\ell > 0$
Multiquadratique	$\left(1 + \dfrac{\|\mathbf{x} - \mathbf{c}^{(i)}\|}{\ell}\right)^{1/2}$	$\ell > 0$
Multiquadratique inverse	$\left(1 + \dfrac{\|\mathbf{x} - \mathbf{c}^{(i)}\|}{\ell}\right)^{-1/2}$	$\ell > 0$

TABLE 3.4 – Quelques fonctions de base radiales



(a) Multiquadratique

(b) Multiquadratique inverse



(c) Gaussienne

FIGURE 3.8 – Exemples de fonctions de base radiales pour 3 valeurs du paramètre ℓ

Dans ses travaux, Orr propose une « mise à l'échelle » permettant de ne plus travailler avec des fonctions radiales de $\|\mathbf{x}-\mathbf{c}\| = (\mathbf{x}-\mathbf{c})^T(\mathbf{x}-\mathbf{c})$ mais de $(\mathbf{x}-\mathbf{c})^T\mathbf{R}(\mathbf{x}-\mathbf{c}) = \sqrt{\sum_{i=1}^{n_p}\frac{(x_i-c_i)^2}{\ell_i^2}}$ [ORR 1999]. Cette modification permet de tenir compte des paramètres du métamodèle différents selon les dimensions et par conséquent de parvenir à une meilleure approximation de fonctions fortement anisotropes. Suite à cette modification, Orr conseille l'emploi de la fonction gaussienne définie de la manière suivante :

$$\phi(\mathbf{x},\mathbf{c}) = \exp\left(-\frac{1}{2}\sum_{i=1}^{nd}\left(\frac{x_i-c_i}{\ell_i}\right)^2\right) \tag{3.62}$$

2.2.3 Détermination des paramètres

Un certain nombre de règles sont présentes dans la littérature permettant de fixer la valeur des paramètres. Ces règles empiriques cherchent à proposer une valeur de paramètre indépendante de la taille du problème (nombre d'échantillons n_s et nombre de variables n_p) et de la nature du problème considéré. Parmi ces valeurs, on trouve par exemple le choix proposé par [HARDY 1971] où $\ell = 1/(0.815d)$ (d est la distance minimale moyenne calculée de la manière proposée par l'équation 2.11 page 113), le choix de [FRANKE 1982] où $\ell = 0.8\sqrt{n_s}/d$ (où d est le diamètre du cercle circonscrit aux données dans l'espace des paramètres), celui de [HAYKIN 1994] proposant de fixer $\ell = d/\sqrt{2n_s}$ (d est la distance maximale entre 2 centres).

Ces suggestions de valeurs, bien qu'elles tiennent compte de la dimension, ne tiennent pas compte de la nature du problème. C'est pourquoi l'emploi d'une méthode plus systématique s'avère indispensable. La solution consiste à chercher le jeu de paramètres du modèle qui minimise un critère basé sur la validation croisée. Cette approche, également utilisée dans certains cas pour le krigeage (cf. partie 2.1.7 page 151), nécessite l'emploi d'une méthode d'optimisation adaptée permettant de localiser de manière efficace le minimum globale du critère. Pour cela, [RIPPA 1999] propose une évaluation efficace du critère de validation croisée et l'emploi d'un algorithme de minimisation adapté (une extension de cette méthode peut être trouvée dans [FASSHAUER et ZHANG 2007]). C. Praveen et R. Duvigneau proposent quant à eux l'emploi d'un algorithme d'optimisation globale de type essaims particulaires [PRAVEEN et DUVIGNEAU 2007].

2.2.4 Variance de prédiction des RBF

En considérant que les réponses y_i utilisées pour construire le modèle RBF sont les réalisations d'un processus stochastique, il est possible d'obtenir la variance de l'erreur statistique du modèle RBF [GIBBS 1997 ; SÓBESTER, LEARY et KEANE 2005] :

$$\boxed{\forall \mathbf{x}^{(0)} \in \mathcal{D},\ s^2_{RBF} = 1 - \boldsymbol{\phi}\big(\mathbf{x}^{(0)}\big)^T\boldsymbol{\Phi}^{-1}\boldsymbol{\phi}\big(\mathbf{x}^{(0)}\big)} \tag{3.63}$$

2.2.5 Version étendue des RBF

Le choix de certaines fonctions de base radiales engendrant une perte de définie-positivité de la matrice $\mathbf{\Phi}$, une version étendue des RBF (*augmented RBF*) a été développée pour corriger cet aspect et assurer l'unicité des coefficients w. Celle-ci consiste à chercher l'approximation sous la forme [SCHABACK 1997] :

$$\tilde{y}\big(\mathbf{x}^{(0)}\big) = \sum_{i=1}^{n_s} w_i \phi\big(\mathbf{x}^{(0)}, \mathbf{c}^{(i)}\big) + \sum_{i=1}^{p} \beta_i f_i\big(\mathbf{x}^{(0)}\big) \tag{3.64}$$

Cette approche ne sera pas employée dans cette thèse.

3 Modèles de substitution à gradients

Comme nous l'avons vu dans le chapitre précédent, il est possible d'obtenir à coût réduit les gradients de la fonction objectif pour un jeu de paramètres donné. L'idée de cette partie est de tenir compte de ces informations pour construire une approximation de la fonction objectif de meilleure qualité. Pour cela, on se concentrera prioritairement sur des modèles de substitution de type krigeage et RBF. On s'intéressera uniquement aux modèles capables de prendre en compte les réponses et toutes les dérivées données en un même point. Dans le cas du krigeage, une extension capable de prendre en compte des données incomplètes est tout à fait envisageable mais ne sera pas abordée (réponse en un point mais pas le gradient ou gradient incomplet).

3.1 Cokrigeage à gradients

La prise en compte de l'information gradient au sein d'un métamodèle de krigeage peut être réalisée de deux manières :

- la première approche consiste à construire un modèle basé sur les développements réalisés dans le cadre de la géostatistique multivariable où par le biais d'un modèle à plusieurs niveaux (un pour les réponses et n_p niveaux pour les n_p dérivées premières). Cette approche, connue sous le nom de **cokrigeage** [WACKERNAGEL 1995], nécessite la mise en œuvre d'une structure de covariance plus évoluée que pour le krigeage permettant d'assurer un lien entre les différents niveaux de l'approche.

- la seconde, plus simple, consiste à extrapoler la valeur de la fonction à approcher en employant un développement de Taylor autour du point échantillonné. La méthode consiste alors à construire un modèle de substitution de type krigeage avec les réponses fournies aux points échantillonnés et celles extrapolées. Cette démarche sera désignée sous le terme **cokrigeage indirect**.

Les développements relatifs au cokrigeage dans un contexte géostatistique existent depuis les années 1970 [MATHERON 1970]. De nombreux travaux visant à étudier et

développer ce modèle se poursuivirent dans les années 1980 et 1990. On pourra se référer aux travaux [ISAAKS et SRIVASTAVA 1989 ; HOEF et CRESSIE 1993 ; GOOVAERTS 1997 ; ARNAUD et EMERY 2000] qui présentent les principes, formulations et discutent des différentes hypothèses nécessaires à l'élaboration du cokrigeage pour la géostatistique.

Bien que s'appuyant sur des fondements du début des années 1990 (en particulier [M. D. MORRIS, MITCHELL et YLVISAKER 1993]), l'emploi du krigeage enrichi avec l'information gradient est relativement récent. Le tableau 3.5 présente un ensemble de références concernant l'emploi du cokrigeage à gradients et des applications associées dans le cadre de l'optimisation ou de l'approximation de fonctions.

Comme l'indique ce tableau, le cokrigeage à gradients a été utilisé à plusieurs reprises dans le contexte de l'optimisation ou de la propagation d'incertitudes portant majoritairement sur des problèmes fluides. Cette situation se justifie principalement par le fait que l'obtention des gradients est réalisée de manière plus aisée sur un problème fluide par approche adjointe. De manière générale, les problèmes fluides, souvent coûteux, sont dans les premiers à avoir nécessité l'emploi de modèles de substitution. En complément de ces différentes références, on pourra également se référer à [ZIMMERMANN 2013] pour des éléments de comparaison entre approche directe et approche indirecte du cokrigeage à gradients.

3.1.1 Formulation du cokrigeage à gradients

L'approche du cokrigeage à gradients s'appuie sur une approche similaire à celle présentée pour l'élaboration du krigeage (cf. partie 2.1.1 page 135). L'idée consiste à considérer une variable dite primaire (la réponse) et des variables auxiliaires (les gradients) et à leur associer des modèles probabilistes sur le même principe que pour le krigeage [WACKERNAGEL 1995] :

$$Y(\mathbf{x}) = \mu_0(\mathbf{x}) + Z_0(\mathbf{x}) \tag{3.65}$$

$$\forall i \in [\![1, n_p]\!],\ W^i(\mathbf{x}) = \mu_i(\mathbf{x}) + Z_i(\mathbf{x}) \tag{3.66}$$

En posant pour simplifier les notations,

$$\forall i \in [\![1, n_p]\!],\ W^i(\mathbf{x}) = \frac{\partial Y}{\partial x_i}(\mathbf{x}) \tag{3.67}$$

On considère pour chaque modèle des propriétés similaires à celle du krigeage. C'est-à-dire que les terme μ_i sont déterministes et les parties Z_i sont aléatoires à moyenne nulle :

$$\mathbb{E}[Y(\mathbf{x})] = \mathbb{E}[\mu_0(\mathbf{x}) + Z_0(\mathbf{x})] = \mu_0(\mathbf{x}) \tag{3.68}$$

$$\sigma_Y^2 = \text{Var}[Y(\mathbf{x})] = \text{Var}[\mu_0(\mathbf{x}) + Z_0(\mathbf{x})] = \text{Var}[Z_0(\mathbf{x})] = \sigma_{Z_0}^2 \tag{3.69}$$

$$\forall i \in [\![1, n_p]\!],\ \mathbb{E}[Z_i(\mathbf{x})] = 0,\ \mathbb{E}\big[W^i(\mathbf{x})\big] = \mu_i(\mathbf{x}) \tag{3.70}$$

$$\sigma_{W^i}^2 = \text{Var}\big[W^i(\mathbf{x})\big] = \text{Var}[\mu_i(\mathbf{x}) + Z_i(\mathbf{x})] = \text{Var}[Z_i(\mathbf{x})] = \sigma_{Z_i}^2 \tag{3.71}$$

Référence	Contenu	Vocabulaire
[M. D. MORRIS, MITCHELL et YLVISAKER 1993]	Développements du cadre de la prise en compte des gradients, applications dans le cadre d'une étude de débit d'un forage	*Bayesian prediction using derivatives, Gaussian Process*
[KOEHLER et OWEN 1996]	Cadre de l'approximation de fonctions, dont le cokrigeage à gradients	*Kriging to model gradients*
[R. M. LEWIS 1998]	Développements théoriques succincts et application sur une fonction analytique	*First Order Kriging*
[CHUNG et ALONSO 2002a ; CHUNG et ALONSO 2002b]	Construction d'un modèle de cokrigeage à gradients pour la minimisation du bruit et de la traînée d'un jet supersonique	*Direct and indirect Cokriging*
[LEARY, BHASKAR et KEANE 2004a]	Utilisation du cokrigeage à gradients pour l'optimisation avec enrichissement, application sur une fonction analytique et optimisation de structure	*Kriging model including derivative information*
[LEARY, BHASKAR et KEANE 2004b]	Comparaison avec d'autres modèles enrichis avec les gradients (RBF, fonction de Shepard) application sur un exemple d'optimisation fluide	*Kriging, gaussian process including derivative*
[LAURENCEAU et SAGAUT 2008]	Application à l'optimisation aérodynamique	*Cokriging, Gradient-Enhanced Kriging*
[LAURENCEAU et MEAUX 2008]	Comparaison au krigeage sans gradient, enrichissement du métamodèle, application à l'optimisation aérodynamique	*Cokriging*
[DWIGHT et Z.-H. HAN 2009]	Application dans un contexte de propagation d'incertitudes, application sur un exemple analytique et fluide	*Gradient-Enhanced Kriging (GEK)*
[XUAN et al. 2009]	Application à l'optimisation de forme aérodynamique	*Gradient-Based Kriging (GBK)*
[LOCKWOOD et MAVRIPLIS 2010]	Application à la propagation d'incertitudes, études de qualité sur des fonctions analytiques et sur un exemple de conception de centrale nucléaire	*Gradient-Enhanced Universal Kriging (GEUK)*
[MARCH, WILLCOX et Q. WANG 2010]	Application à l'optimisation de structures et aérodynamique	*Cokriging*
[W. YAMAZAKI, RUMPFKEIL et MAVRIPLIS 2010]	Extension à la prise en compte des hessiens, applications sur des fonctions analytiques, exemple aérodynamique et optimisation	*Gradient/Hessian-enhanced Direct/Indirect Kriging (GEK)*
[RUMPFKEIL, W. YAMAZAKI et MAVRIPLIS 2011]	Etude de qualité sur des fonctions analytiques, étude de robustesse sur un profil d'aile transonique et propagation d'incertitudes	*Cokriging, Gradient and Hessian enhanced Kriging*
[Z.-H. HAN, GÖRTZ et ZIMMERMANN 2013]	Approche multifidélité appliquée à l'étude d'un profil d'aile	*direct Gradient-Enhanced Kriging (GEK)*

TABLE 3.5 – Quelques références concernant l'emploi du cokrigeage à gradients

Les modèles de tendance μ_i peuvent être choisis indépendamment les uns des autres [CHAUVET 1999] et conduisent à différents types de cokrigeage (simple, ordinaire ou universel). Pour alléger les développements de cette partie, nous choisirons de développer le cokrigeage universel c'est-à-dire tel que :

$$\mu_0(\mathbf{x}) = \sum_{i=1}^{p} \beta_i f_i(\mathbf{x}) = \mathbf{f}_0^T \boldsymbol{\beta} \tag{3.72}$$

où $\boldsymbol{\beta} = \begin{bmatrix}\beta_1 & \beta_2 & \dots & \beta_p\end{bmatrix}^T$ et $\mathbf{f}_0 = \begin{bmatrix}f_1\big(\mathbf{x}^{(0)}\big) & f_2\big(\mathbf{x}^{(0)}\big) & \dots & f_p\big(\mathbf{x}^{(0)}\big)\end{bmatrix}^T$.

Par ailleurs, nous limiterons les développements au cas où les termes de tendance μ_i sont déduits du terme μ_0 par dérivation (voir [M. D. MORRIS, MITCHELL et YLVISAKER 1993]). Ainsi,

$$\forall i \in [\![1, n_p]\!],\ \mu_i(\mathbf{x}) = \frac{\partial \mu_0}{\partial x_i}(\mathbf{x}) = \sum_{j=1}^{p} \beta_j \frac{\partial f_j}{\partial x_i}(\mathbf{x}) = \mathbf{f}_0^{iT} \boldsymbol{\beta} \tag{3.73}$$

où $\forall i \in [\![1, n_p]\!],\ \mathbf{f}_0^i = \begin{bmatrix}\dfrac{\partial f_1}{\partial x_i}(\mathbf{x}^{(0)}) & \dfrac{\partial f_2}{\partial x_i}(\mathbf{x}^{(0)}) & \dots & \dfrac{\partial f_p}{\partial x_i}(\mathbf{x}^{(0)})\end{bmatrix}^T$.

L'estimateur du cokrigeage est ensuite recherché en un point quelconque $\mathbf{x}^{(0)}$ sous la forme d'une combinaison linéaire de l'ensemble des variables :

$$\boxed{\forall \mathbf{x}^{(0)} \in \mathcal{D},\ \hat{Y}\big(\mathbf{x}^{(0)}\big) = \sum_{i=1}^{n_s} \lambda_i^0(\mathbf{x}^{(0)}) Y\big(\mathbf{x}^{(i)}\big) + \sum_{i=1}^{n_s}\sum_{j=1}^{n_p} \lambda_i^j(\mathbf{x}^{(0)}) W^j\big(\mathbf{x}^{(i)}\big)} \tag{3.74}$$

A l'instar du krigeage, la méthode consiste ici à minimiser la variance de l'erreur d'estimation sous la condition de non-biais.

■ **Condition de non biais :** La condition de non-biais se formule de la manière suivante :

$$\begin{aligned}
\forall \mathbf{x}^{(0)} \in \mathcal{D},\ \mathbb{E}\Big[\hat{Y}\big(\mathbf{x}^{(0)}\big) - Y\big(\mathbf{x}^{(0)}\big)\Big] &= 0 \\
\mathbb{E}\left[\sum_{i=1}^{n_s} \lambda_i^0(\mathbf{x}^{(0)}) Y\big(\mathbf{x}^{(i)}\big) + \sum_{i=1}^{n_p}\sum_{j=1}^{n_p} \lambda_i^j(\mathbf{x}^{(0)}) W^j\big(\mathbf{x}^{(i)}\big) - Y\big(\mathbf{x}^{(0)}\big)\right] &= 0 \\
\sum_{i=1}^{n_s} \lambda_i^0(\mathbf{x}^{(0)}) \mathbb{E}\big[Y\big(\mathbf{x}^{(i)}\big)\big] + \sum_{i=1}^{n_s}\sum_{j=1}^{n_p} \lambda_i^j(\mathbf{x}^{(0)}) \mathbb{E}\big[W^j\big(\mathbf{x}^{(i)}\big)\big] - \mathbb{E}\big[Y\big(\mathbf{x}^{(0)}\big)\big] &= 0 \\
\sum_{i=1}^{n_s} \lambda_i^0(\mathbf{x}^{(0)}) \mu_0\big(\mathbf{x}^{(i)}\big) + \sum_{i=1}^{n_s}\sum_{j=1}^{n_p} \lambda_i^j(\mathbf{x}^{(0)}) \mu_j\big(\mathbf{x}^{(i)}\big) - \mu_0\big(\mathbf{x}^{(0)}\big) &= 0
\end{aligned} \tag{3.75}$$

Les développements choisis pour les modèles de tendance présentés par les équations 3.72 et 3.73 conduisent à la formulation de la condition de non-biais (équation 3.75)

sous la forme :

$$\sum_{i=1}^{n_s} \lambda_i^0(\mathbf{x}^{(0)})\mu_0\big(\mathbf{x}^{(i)}\big) + \sum_{i=1}^{n_s}\sum_{j=1}^{n_p} \lambda_i^j(\mathbf{x}^{(0)})\mu_j\big(\mathbf{x}^{(i)}\big) - \mu_0\big(\mathbf{x}^{(0)}\big) = 0$$
$$\sum_{i=1}^{n_s} \lambda_i^0(\mathbf{x}^{(0)}) \sum_{k=1}^{p} \beta_k f_k\big(\mathbf{x}^{(i)}\big) + \sum_{i=1}^{n_s}\sum_{j=1}^{n_p} \lambda_i^j(\mathbf{x}^{(0)}) \sum_{k=1}^{p} \beta_k \frac{\partial f_k}{\partial x_j}(\mathbf{x}^{(i)}) - \sum_{k=1}^{p} \beta_k f_k\big(\mathbf{x}^{(0)}\big) = 0 \tag{3.76}$$

On pose

$$\boldsymbol{\lambda}_0 = \begin{bmatrix}\lambda_1^0 & \lambda_2^0 & \dots & \lambda_{n_s}^0\end{bmatrix}^T \quad \text{de taille } 1 \times n_s$$
$$\boldsymbol{\lambda}_W = \begin{bmatrix}\lambda_1^1 & \lambda_1^2 & \dots & \lambda_1^{n_p} & \lambda_2^1 & \dots & \lambda_{n_s}^{n_p}\end{bmatrix}^T \quad \text{de taille } 1 \times n_s n_p$$
$$\mathbf{F} = \begin{bmatrix} f_1\big(\mathbf{x}^{(1)}\big) & f_2\big(\mathbf{x}^{(1)}\big) & \dots & f_p\big(\mathbf{x}^{(1)}\big) \\ f_1\big(\mathbf{x}^{(2)}\big) & \ddots & & \\ \vdots & & & \\ f_1\big(\mathbf{x}^{(n_s)}\big) & \dots & \dots & f_p\big(\mathbf{x}^{(n_s)}\big) \end{bmatrix} \quad \text{de taille } p \times n_s$$
$$\mathbf{F}_W = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}^{(1)}) & \frac{\partial f_2}{\partial x_1}(\mathbf{x}^{(1)}) & \dots & \frac{\partial f_p}{\partial x_1}(\mathbf{x}^{(1)}) \\ \vdots & & & \vdots \\ \frac{\partial f_1}{\partial x_{n_p}}(\mathbf{x}^{(1)}) & \frac{\partial f_2}{\partial x_{n_p}}(\mathbf{x}^{(1)}) & \dots & \frac{\partial f_p}{\partial x_{n_p}}(\mathbf{x}^{(1)}) \\ \frac{\partial f_1}{\partial x_1}(\mathbf{x}^{(2)}) & \frac{\partial f_2}{\partial x_1}(\mathbf{x}^{(2)}) & \dots & \frac{\partial f_p}{\partial x_1}(\mathbf{x}^{(2)}) \\ \vdots & & & \vdots \\ \frac{\partial f_1}{\partial x_{n_p}}(\mathbf{x}^{(n_s)}) & \frac{\partial f_2}{\partial x_{n_p}}(\mathbf{x}^{(n_s)}) & \dots & \frac{\partial f_p}{\partial x_{n_p}}(\mathbf{x}^{(n_s)}) \end{bmatrix} \quad \text{de taille } p \times n_s n_p$$

L'équation 3.76 s'écrit alors

$$\boldsymbol{\lambda}_0^T \mathbf{F}\boldsymbol{\beta} + \boldsymbol{\lambda}_W^T \mathbf{F}_W \boldsymbol{\beta} = \mathbf{f}_0^T \boldsymbol{\beta} \tag{3.77}$$

Soit dans une version matricielle condensée et en supposant le modèle de tendance non nul, la condition de non-biais se résume à :

$$\boldsymbol{\lambda}_c \mathbf{F}_c = \mathbf{f}_0^T \tag{3.78}$$

où le vecteur $\boldsymbol{\lambda}_c = \begin{bmatrix}\boldsymbol{\lambda}_0^T & \boldsymbol{\lambda}_W^T\end{bmatrix}^T$ contient les $(n_p + 1) \times n_s$ coefficients du cokrigeage et $\mathbf{F}_c = \begin{bmatrix}\mathbf{F}^T & \mathbf{F}_W^T\end{bmatrix}^T$ est une matrice comportant $(n_s + 1) \times n_s$ lignes et p colonnes.

■ **Formulation de la variance :** La variance de l'erreur d'estimation du cokrigeage s'écrit de la manière suivante :

$$
\begin{aligned}
&\forall \mathbf{x}^{(0)} \in \mathcal{D}, \\
s^2_{CKU}(\mathbf{x}^{(0)}) &= \mathrm{Var}\Big[\hat{Y}\big(\mathbf{x}^{(0)}\big) - Y\big(\mathbf{x}^{(0)}\big)\Big] \\
&= \mathrm{Var}\Big[\hat{Y}\big(\mathbf{x}^{(0)}\big)\Big] + \mathrm{Var}\big[Y\big(\mathbf{x}^{(0)}\big)\big] - 2\mathrm{cov}\Big[\hat{Y}\big(\mathbf{x}^{(0)}\big), Y\big(\mathbf{x}^{(0)}\big)\Big] \\
&= \mathrm{Var}\left[\sum_{i=1}^{n_s} \lambda_i^0(\mathbf{x}^{(0)}) Z_0\big(\mathbf{x}^{(i)}\big) + \sum_{i=1}^{n_s}\sum_{j=1}^{n_p} \lambda_i^j(\mathbf{x}^{(0)}) Z_j\big(\mathbf{x}^{(i)}\big)\right] \\
&+ \mathrm{Var}\big[Z_0\big(\mathbf{x}^{(0)}\big)\big] \\
&- 2\mathrm{cov}\left[\sum_{i=1}^{n_s} \lambda_i^0(\mathbf{x}^{(0)}) Z_0\big(\mathbf{x}^{(i)}\big) + \sum_{i=1}^{n_s}\sum_{j=1}^{n_p} \lambda_i^j(\mathbf{x}^{(0)}) Z_j\big(\mathbf{x}^{(i)}\big), Z_0\big(\mathbf{x}^{(0)}\big)\right] \\
&= \sigma^2_{Z_0} + \sum_{i=1}^{n_s}\sum_{j=1}^{n_p} \lambda_i^0\lambda_j^0 \mathrm{cov}\big[Z_0\big(\mathbf{x}^{(i)}\big), Z_0\big(\mathbf{x}^{(j)}\big)\big] \\
&+ \sum_{i=1}^{n_s}\sum_{k=1}^{n_s}\sum_{j=1}^{n_p}\sum_{l=1}^{n_p} \lambda_i^j\lambda_j^l \mathrm{cov}\big[Z_j\big(\mathbf{x}^{(i)}\big), Z_l\big(\mathbf{x}^{(k)}\big)\big] \qquad (3.79) \\
&+ 2\sum_{i=1}^{n_s}\sum_{k=1}^{n_s}\sum_{j=1}^{n_p} \lambda_i^0\lambda_k^j \mathrm{cov}\big[Z_0\big(\mathbf{x}^{(i)}\big), Z_j\big(\mathbf{x}^{(k)}\big)\big] \qquad (3.80) \\
&- 2\sum_{i=1}^{n_s}\sum_{j=1}^{n_p} \lambda_i^j \mathrm{cov}\big[Z_j\big(\mathbf{x}^{(i)}\big), Z_0\big(\mathbf{x}^{(0)}\big)\big] - 2\sum_{i=1}^{n_s} \lambda_i^j \mathrm{cov}\big[Z_0\big(\mathbf{x}^{(i)}\big), Z_0\big(\mathbf{x}^{(0)}\big)\big]
\end{aligned}
$$

Afin de faciliter la formulation matricielle de la variance du cokrigeage universel, on pose, pour les covariances et covariances croisées :

$$
\begin{aligned}
&\forall (i,j,k,l) \in [\![0, n_s]\!]^2 \times [\![1, n_s]\!]^2, \\
&\mathrm{cov}\big[Z_0\big(\mathbf{x}^{(i)}\big), Z_0\big(\mathbf{x}^{(j)}\big)\big] = \mathrm{cov}\big[Y\big(\mathbf{x}^{(i)}\big), Y\big(\mathbf{x}^{(j)}\big)\big] = c_{ij} \\
&\mathrm{cov}\big[Z_0\big(\mathbf{x}^{(i)}\big), Z_k\big(\mathbf{x}^{(j)}\big)\big] = \mathrm{cov}\big[Y\big(\mathbf{x}^{(i)}\big), W^k\big(\mathbf{x}^{(j)}\big)\big] = c_{ij,k} \\
&\mathrm{cov}\big[Z_k\big(\mathbf{x}^{(i)}\big), Z_l\big(\mathbf{x}^{(j)}\big)\big] = \mathrm{cov}\big[W^k\big(\mathbf{x}^{(i)}\big), W^l\big(\mathbf{x}^{(j)}\big)\big] = c_{ij,kl}
\end{aligned} \qquad (3.81)
$$

Ainsi, la variance du cokrigeage s'écrit sous forme matricielle :

$$
\forall \mathbf{x}^{(0)} \in \mathcal{D},\ s^2_{CKU}(\mathbf{x}^{(0)}) = \sigma^2_{Z_0} + \boldsymbol{\lambda}_c^T \mathbf{C}_c \boldsymbol{\lambda}_c - 2\boldsymbol{\lambda}_c^T \mathbf{c}_{0c} \qquad (3.82)
$$

où la matrice des covariances $\mathbf{C}_c = \begin{bmatrix} \mathbf{C} & \mathbf{C}_{WY} \\ \mathbf{C}_{WY}^T & \mathbf{C}_{WW} \end{bmatrix}$ est constituée de la matrice de covariance employée pour le krigeage $\mathbf{C}$, la matrice de covariance $\mathbf{C}_{WY}$ liant les variables primaires et auxiliaires ainsi que celle liant les variables auxiliaires $\mathbf{C}_{WW}$. Ces matrices sont

construites de la manière suivante, en employant les notations données par l'équation 3.81 :

$$(\mathbf{C})_{ij} = c_{ij}$$

$$\mathbf{C}_{WY} = \begin{bmatrix} c_{11,1} & c_{11,2} & \cdots & c_{11,n_p} & c_{12,1} & \cdots & c_{1n_s,n_p} \\ c_{21,1} & c_{21,2} & \cdots & c_{21,n_p} & c_{22,1} & \cdots & c_{2n_s,n_p} \\ c_{31,1} & c_{31,2} & \cdots & & & & \\ \vdots & & \ddots & & & & \vdots \\ c_{n_s1,1} & \cdots & & & & & c_{n_sn_s,n_p} \end{bmatrix} \text{ de taille } n_s n_p \times n_s$$

$$\mathbf{C}_{WW} = \begin{bmatrix} \mathbf{C}_{WW}^{11} & \mathbf{C}_{WW}^{12} & \cdots & \mathbf{C}_{WW}^{1n_s} \\ \mathbf{C}_{WW}^{21} & \mathbf{C}_{WW}^{22} & \cdots & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{C}_{WW}^{n_s1} & \mathbf{C}_{WW}^{n_s2} & \cdots & \mathbf{C}_{WW}^{n_sn_s} \end{bmatrix} \text{ de taille } n_s n_p \times n_s n_p$$

$$\text{avec } \forall (k,l) \in [\![1, n_s]\!]^2,\ \mathbf{C}_{WW}^{kl} = \begin{bmatrix} c_{kl,11} & c_{kl,12} & \cdots & c_{kl,1n_p} \\ c_{kl,21} & c_{kl,22} & \cdots & \vdots \\ \vdots & & \ddots & \vdots \\ c_{kl,n_p1} & c_{kl,n_p2} & \cdots & c_{kl,n_pn_p} \end{bmatrix}$$

Au bilan, la matrice des covariances $\mathbf{C}_c$ obtenue est symétrique et comporte $n_s(n_p+1)$ lignes et autant de colonnes.

■ **Formulation du problème de cokrigeage universel :** Ainsi décrite à partir des formulations de la variance du cokrigeage (équation 3.82) et de la condition de non-biais (équation 3.78), l'obtention du modèle de cokrigeage repose sur la résolution du problème suivant :

Problème 3.4 (Cokrigeage Universel)
Trouver $\boldsymbol{\lambda}_c \in \mathbb{R}^{n_s(n_p+1)}$ minimisant

$$\boldsymbol{\lambda}_c^T \mathbf{C}_c \boldsymbol{\lambda}_c - 2\boldsymbol{\lambda}_c^T \mathbf{c}_{0c} + \sigma_{Z_0}^2$$

sous la contrainte $\mathbf{F}_c^T \boldsymbol{\lambda}_c = \mathbf{f}_0$

Ce problème est du même type que celui rencontré lors de l'élaboration du krigeage universel (problème 3.3 page 141). La résolution est réalisée de la même manière et on obtient l'approximation par cokrigeage universel ainsi que la variance suivante :

$$\forall \mathbf{x}^{(0)} \in \mathcal{D},\quad \tilde{y}_{CKU}\left(\mathbf{x}^{(0)}\right) = \left(\mathbf{c}_{0c} + \mathbf{F}_c \left(\mathbf{F}_c^T \mathbf{C}_c^{-1} \mathbf{F}_c\right)^{-1} \left(\mathbf{f}_0 - \mathbf{F}_c^T \mathbf{C}_c^{-1} \mathbf{c}_{0c}\right)\right)^T \mathbf{C}_c^{-1} \mathbf{y}_c \quad (3.83)$$

avec $\mathbf{y}_c = \begin{bmatrix} y_1 & \dots & y_{n_s} & \dfrac{\partial y_1}{\partial x_1} & \dfrac{\partial y_1}{\partial x_2} & \dots & \dfrac{\partial y_{n_s}}{\partial x_{n_p}} \end{bmatrix}^T$.

$$\boxed{\begin{array}{c} \forall \mathbf{x}^{(0)} \in \mathcal{D},\ s^2_{CKU}\big(\mathbf{x}^{(0)}\big) = \sigma^2_{Z_0} - \mathbf{c}_{0c}{}^T \mathbf{C}_c^{-1} \mathbf{c}_{0c} + \mathbf{u}_0^T \left(\mathbf{F}_c^T \mathbf{C}_c^{-1} \mathbf{F}_c\right)^{-1} \mathbf{u}_0 \\ \text{avec } \mathbf{u}_0 = \mathbf{u}\big(\mathbf{x}^{(0)}\big) = \mathbf{F}_c^T \mathbf{C}_c^{-1} \mathbf{c}_{0c} - \mathbf{f}_0 \end{array}} \tag{3.84}$$

Remarque 3.12 : De la même manière que pour le krigeage universel on peut déduire du cokrigeage universel, le cokrigeage ordinaire (CKO) en considérant que $\mu_0(\mathbf{x}) = \beta$ et que d'après l'équation 3.73, $\forall i \in [\![1, n_p]\!]$, $\mu_i(\mathbf{x}) = 0$. Dans ce cas, $\mathbf{F}_c = \begin{bmatrix} \mathbf{1}_{n_s}^T & \mathbf{0}_{n_s \times n_p}^T \end{bmatrix}^T$ où $\mathbf{1}_{n_s}$ et $\mathbf{0}_{n_s \times n_p}$ sont des vecteurs de dimensions respectives n_s et $n_s \times n_p$.

■ **Structure de covariance :** La difficulté principale dans l'emploi du cokrigeage réside dans le choix d'une structure de covariance. En effet, les covariances croisées (entre variable primaire et variables auxiliaires) peuvent s'avérer complexes à établir. Cependant, dans le cas de la prise en compte des gradients, ces relations sont clairement établies et proposées dans [M. D. MORRIS, MITCHELL et YLVISAKER 1993] sous la forme générale :

$$\begin{array}{l} \forall k \in [\![0, n_p]\!] \forall (a_k, b_k) \in \mathbb{N}^2,\ \forall (\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \in \mathcal{D}, \\ \operatorname{cov}\left[Y^{(a_1,a_2,\dots,a_{n_p})}\big(\mathbf{x}^{(i)}\big), Y^{(b_1,b_2,\dots,b_{n_p})}\big(\mathbf{x}^{(j)}\big)\right] = \sigma_Y^2 (-1)^{\sum a_j} \displaystyle\prod_{j=1}^{n_p} h^{(a_j+b_j)}(x_j^{(i)} - x_j^{(j)}; \ell_j) \end{array} \tag{3.85}$$

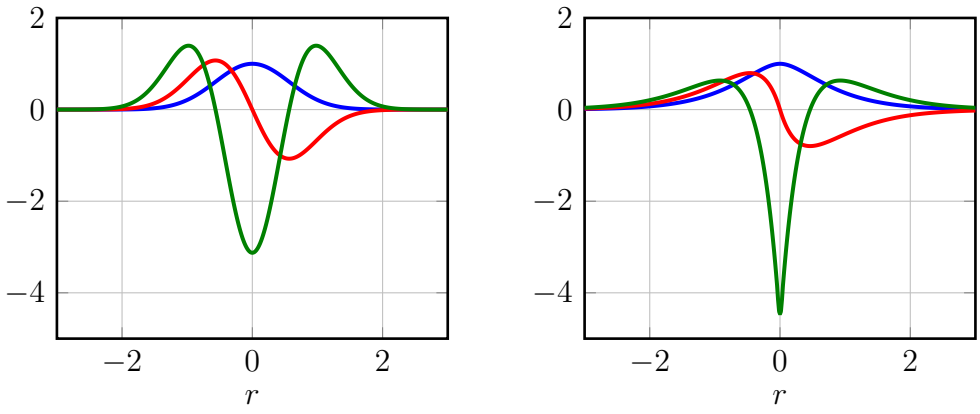
avec

$$Y^{(a_1,a_2,\dots,a_{n_p})}\big(\mathbf{x}^{(i)}\big) = \frac{\partial^{a_1+a_2\cdots+a_{n_p}} Y}{\partial x_1^{a_1} \partial x_2^{a_2} \dots \partial x_{n_p}^{a_{n_p}}}$$

Dans le cas du cokrigeage à gradients, il est nécessaire de considérer les relations suivantes (équation 3.86) :

$$\begin{array}{l} \forall (i,j,k,l) \in [\![0, n_s]\!]^2 \times [\![1, n_s]\!]^2, \\ \qquad \operatorname{cov}\left[Y\big(\mathbf{x}^{(i)}\big), Y\big(\mathbf{x}^{(j)}\big)\right] = c_{ij} = c_{ji} = \sigma_Y^2 R\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \mathbf{l}\big) \\ \qquad \operatorname{cov}\left[Y\big(\mathbf{x}^{(i)}\big), \dfrac{\partial Y}{\partial x_k}\big(\mathbf{x}^{(j)}\big)\right] = c_{ij,k} = \sigma_Y^2 \dfrac{\partial R}{\partial x_k}\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \mathbf{l}\big) \\ \qquad \operatorname{cov}\left[\dfrac{\partial Y}{\partial x_k}\big(\mathbf{x}^{(j)}\big), Y\big(\mathbf{x}^{(i)}\big)\right] = c_{ji,k} = -\sigma_Y^2 \dfrac{\partial R}{\partial x_k}\big(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}; \mathbf{l}\big) \\ \qquad \operatorname{cov}\left[\dfrac{\partial Y}{\partial x_k}\big(\mathbf{x}^{(i)}\big), \dfrac{\partial Y}{\partial x_l}\big(\mathbf{x}^{(j)}\big)\right] = c_{klij} = -\sigma_Y^2 \dfrac{\partial^2 R}{\partial x_k \partial x_l}\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \mathbf{l}\big) \end{array} \tag{3.86}$$

Les fonctions de corrélation sont des fonctions paires mais leurs dérivées premières sont impaires, c'est pourquoi on retrouve le résultat de l'équation 3.81 où $\forall (i,j,k) \in [\![1, n_s]\!]^2 \times [\![1, n_p]\!],\ c_{ij,k} = -c_{ji,k}$.



(a) Fonction Gaussienne **(b)** Fonction Matérn $\nu = 3/2$

FIGURE 3.9 – Exemples de fonctions de corrélations Gaussienne et Matérn (valeur de la fonction, dérivées premières et dérivées secondes pour $\ell = 0.8$)

Remarque 3.13 : Par ailleurs, ces résultats de covariance permettent de formuler l'estimateur et la variance du cokrigeage à gradients sous une forme faisant intervenir des matrices et vecteurs de corrélation comme ceux présentés dans le tableau 3.3 page 146.

Remarque 3.14 : Les relations de covariances issues de [M. D. MORRIS, MITCHELL et YLVISAKER 1993] établissent également que $\forall i \in [\![1, n_p]\!]$, $\sigma_Y^2 = \sigma_{Z_0}^2 = \sigma_{W^i}^2 = \sigma_{Z_i}^2$

Dans le cas multidimensionnel ($n_p > 1$) où la fonction de corrélation est définie telle que $R\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \mathbf{l}\big) = \prod_{k=1}^{n_p} h(x_k^{(i)} - x_k^{(j)}; \ell_k)$, le calcul des dérivées de la fonction de corrélation conduit aux équations :

$$\frac{\partial R}{\partial x_m}\big(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}; \mathbf{l}\big) = \frac{\mathrm{d}h}{\mathrm{d}x_m}\big(x_m^{(i)} - x_m^{(j)}; \ell_m\big) \prod_{\substack{k=1 \\ k\neq m}}^{n_p} h(x_k^{(i)} - x_k^{(j)}; \ell_k)$$

$$\frac{\partial^2 R}{\partial x_m \partial x_n}\big(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}; \mathbf{l}\big) = \frac{\mathrm{d}h}{\mathrm{d}x_m}\big(x_m^{(i)} - x_m^{(j)}; \ell_m\big) \frac{\mathrm{d}h}{\mathrm{d}x_n}\big(x_n^{(i)} - x_n^{(j)}; \ell_n\big) \prod_{\substack{k=1 \\ k\neq m,n}}^{n_p} h(x_k^{(i)} - x_k^{(j)}; \ell_k)$$

$$\frac{\partial^2 R}{\partial {x_m}^2}\big(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}; \mathbf{l}\big) = \frac{\mathrm{d}^2 h}{\mathrm{d}{x_m}^2}\big(x_m^{(i)} - x_m^{(j)}; \ell_m\big) \prod_{\substack{k=1 \\ k\neq m}}^{n_p} h(x_k^{(i)} - x_k^{(j)}; \ell_k)$$

D'un point de vue pratique, Lockwood propose une méthode pouvant faciliter l'implémentation et le calcul des dérivées [LOCKWOOD et ANITESCU 2012]. Cette approche consiste

à simplement calculer les 3 grandeurs suivantes :

$$L_m = \prod_{k<m} h(x_k^{(i)} - x_k^{(j)}; \ell_k) \tag{3.87}$$

$$U_m = \prod_{k>m} h(x_k^{(i)} - x_k^{(j)}; \ell_k) \tag{3.88}$$

$$M_{m,n} = \prod_{m<k<n} h(x_k^{(i)} - x_k^{(j)}; \ell_k) \text{ avec } m < n \tag{3.89}$$

Le calcul des dérivées est alors réalisé de la manière suivante :

$$\begin{aligned}
\frac{\partial R}{\partial x_m}\left(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}; \mathbf{l}\right) &= L_m U_m \frac{\mathrm{d}h}{\mathrm{d}x_m} \\
\frac{\partial^2 R}{\partial x_m \partial x_n}\left(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}; \mathbf{l}\right) &= L_m M_{m,n} U_n \frac{\mathrm{d}h}{\mathrm{d}x_m}\left(x_m^{(i)} - x_m^{(j)}; \ell_m\right) \frac{\mathrm{d}h}{\mathrm{d}x_n}\left(x_n^{(i)} - x_n^{(j)}; \ell_n\right) \\
\frac{\partial^2 R}{\partial {x_m}^2}\left(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}; \mathbf{l}\right) &= L_m U_m \frac{\mathrm{d}^2 h}{\mathrm{d}{x_m}^2}\left(x_m^{(i)} - x_m^{(j)}; \ell_m\right)
\end{aligned}$$

L'avantage de cette méthode est qu'elle ne nécessite que le calcul des dérivées des fonctions de corrélation unidimensionnelles h permettant ainsi une implémentation et un contrôle plus simple du calcul des gradients.

■ **Détermination des paramètres du cokrigeage :** A l'instar du krigeage, dans le cas du cokrigeage à gradients, la détermination des paramètres peut être réalisée en employant une des méthodes introduites dans la partie 2.1.7 page 147. Dans le cadre de ces travaux, la méthode utilisée sera celle du maximum de vraisemblance où l'on cherche à maximiser la grandeur suivante :

$$\begin{aligned}
&L(\boldsymbol{\beta}, \sigma_Z^2, \mathbf{l}) = \\
&\left(2\pi\sigma_Z^2\right)^{-n_s(n_p+1)/2} |\mathbf{K}(\mathbf{l})|^{-1/2} \exp\left[-\frac{1}{2\sigma_Z^2}(\mathbf{Y}_{n_s} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{K}(\mathbf{l})^{-1}(\mathbf{Y}_{n_s} - \mathbf{F}\boldsymbol{\beta})\right]
\end{aligned} \tag{3.90}$$

■ **Quelques aspects pratiques :** La construction du cokrigeage à gradients et en particulier les étapes relatives à la construction de la matrice de corrélation (cette construction survient autant de fois que l'algorithme de minimisation de la -log-vraisemblance nécessite une valeur de cette dernière) peuvent devenir coûteuses. En effet, la matrice de corrélation du cokrigeage comporte $n_s^2 \times (n_p + 1)^2$ termes au lieu des n_s^2 termes d'une matrice de corrélation classique pour le krigeage. Le coût de construction peut alors devenir très important pour des problèmes qui comportent un nombre important de paramètres et d'échantillons.

Dans le cadre de ces travaux, où le temps de calcul associé à la résolution du problème mécanique reste prépondérant, peu d'études du temps de construction et d'optimisation de code ont été réalisées pour chercher à réduire le temps de construction du métamodèle. Cependant, l'implémentation sous MATLAB a permis de mettre facilement en œuvre

une parallélisation de la construction de la matrice de corrélation et de la détermination des paramètres. Cette parallélisation « multithreadée » est réalisée par l'intermédiaire de l'opérateur `parfor` et en maximisant les opérations vectorielles et l'utilisation de fonctions parallélisées incluses par défaut dans MATLAB. Les machines utilisées dans le cadre de cette thèse comportant jusqu'à 16 cœurs, d'importants gains de temps pour la construction du krigeage ont pu être obtenus. Une implémentation plus fine et l'emploi d'un code compilé (C++ par exemple) permettraient des gains encore plus importants.

3.1.2 Cokrigeage indirect

Le cokrigeage indirect [W. LIU 2003] est une méthode d'interpolation permettant de tenir compte de l'information gradient obtenue aux points échantillonnés tout en employant un modèle de substitution de type krigeage classique. L'idée repose sur un développement de Taylor à l'ordre 1 au point échantillonné $\mathbf{x}^{(j)}$. Ainsi, pour chacune des n_p variables, un point voisin ${\mathbf{x}^{(j)}}^{vi}$ est construit tel que :

$$\forall (i,j) \in [\![1, n_p]\!] \times [\![1, n_s]\!],\ {\mathbf{x}^{(j)}}^{vi} = \mathbf{x}^{(j)} + \Delta x_i \mathbf{e}_i \tag{3.91}$$

où $\mathbf{e}_i$ est un vecteur de base orthonormale de l'espace des paramètres. On considère que $|\Delta x_i| \ll 1$ et ainsi le développement de Taylor à l'ordre 1 permet d'extrapoler la valeur de la fonction y au point ${\mathbf{x}^{(j)}}^{vi}$:

$$y\Big({\mathbf{x}^{(j)}}^{vi}\Big) \approx y\big(\mathbf{x}^{(j)}\big) + \frac{\partial y}{\partial x_i}\big(\mathbf{x}^{(j)}\big)\Delta x_i \tag{3.92}$$

Dans le cas général, le krigeage est alors construit à partir de $n_s \times (nd + 1)$ points et réponses de la fonction y. Cette méthode nécessite de choisir des valeurs de Δx_i pertinentes : ces paramètres ne doivent pas être choisis trop faibles. Cela entraînerait en effet une altération du conditionnement de la matrice de corrélation du krigeage. Ils ne doivent pas non plus être choisis trop élevés car cela risquerait de fausser l'extrapolation (liée au développement de Taylor) et par conséquent de fournir une approximation erronée. Dans ses travaux, Liu propose de déterminer les paramètres Δx_i par l'intermédiaire du maximum de vraisemblance. [LAURENCEAU et MEAUX 2008] proposent quant à eux de choisir $\forall i \in [\![1, n_p]\!],\ \Delta x = \Delta x_i = 10^{-4}$.

Cette approche présente le défaut d'être moins robuste que l'approche de krigeage classique et peut engendrer d'importants problèmes de conditionnement de la matrice de corrélation en raison de l'ajout de points proches dans l'espace des paramètres.

Remarque 3.15 : On remarquera que cette stratégie peut s'adapter également à d'autres types de modèles de substitution qui ne prennent pas en compte l'information gradient.

3.1.3 Extension à des dérivées d'ordre supérieur

La démarche proposée dans le cadre de la prise en compte des gradients par l'intermédiaire du cokrigeage peut bien entendu être étendue à des dérivées d'ordre supérieur. La complexité du modèle s'en trouve alors augmentée et une attention particulière doit être maintenue sur l'élaboration de la structure de covariance (en particulier le choix de la fonction de corrélation). Ce type de modèle de substitution ne sera pas développé dans ces travaux mais quelques éléments peuvent être trouvés dans [FORRESTER, SÓBESTER et KEANE 2008].

3.2 RBF à gradients

Souvent désignée sous le terme d'interpolation de Hermite par fonctions de base radiales [ZONGMIN 1992], cette méthode propose l'intégration de l'information gradient dans les réseaux de fonctions de bases radiales. Introduite dans le contexte plus vaste des réseaux de neurones [I. C. KAMPOLIS, KARANGELOS et K. C. GIANNAKOGLOU 2004 ; K. GIANNAKOGLOU, PAPADIMITRIOU et I. KAMPOLIS 2006], elle a également été développée dans le cadre de l'approximation de fonctions et pour l'optimisation en dynamique des fluides [LEARY, BHASKAR et KEANE 2004b] et dans celui de l'optimisation aérodynamique d'un profil d'aile d'un point [ONG, LUM et NAIR 2008].

■ **Elaboration :** L'idée des RBF à gradients repose sur l'ajout de nouvelles fonctions de base pour construire l'approximation. Ces nouvelles fonctions sont choisies comme étant les dérivées des fonctions RBF. Ainsi, l'approximation en un point quelconque $\mathbf{x}^{(0)}$ est cherchée sous la forme :

$$\begin{aligned}\tilde{y}\big(\mathbf{x}^{(0)}\big) &= \sum_{i=1}^{n_s} w_i \phi\big(\mathbf{x}^{(0)}, \mathbf{x}^{(i)}\big) + \sum_{j=1}^{n_p}\sum_{i=1}^{n_s} w_{ij} \frac{\partial \phi}{\partial x_j}\big(\mathbf{x}^{(0)}, \mathbf{x}^{(i)}\big) \\ &= \sum_{j=0}^{n_p}\sum_{i=1}^{n_s} w_{ij}\phi_{0i,j} \end{aligned} \tag{3.93}$$

avec $\forall (i,j) \in [\![1, n_s]\!] \times [\![0, n_p]\!],\ w_{ij} = \begin{cases} w_{i0} = w_i & \text{si } j = 0 \\ w_{ij} & \text{sinon} \end{cases}$.

$$\forall \mathbf{x}^{(0)} \in \mathcal{D},\ \forall (i,j,k) \in [\![1, n_s]\!] \times [\![0, n_s]\!] \times [\![0, n_p]\!], \tag{3.94}$$

$$\phi_{ij,k} = \begin{cases} \phi_{ij,0} = \phi_{ij} = \phi\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\big) & \text{si } k = 0 \\ \phi_{ij,k} = \dfrac{\partial \phi_{ij}}{\partial x_k} = \dfrac{\partial \phi}{\partial x_k}\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\big) & \text{sinon} \end{cases} \tag{3.95}$$

On introduit également la notation $\phi_{ij,kl} = \dfrac{\partial^2 \phi}{\partial x_k \partial x_l}\big(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\big)$ pour les dérivées secondes. On considère, comme fonction radiale, la fonction Gaussienne qui est de classe $\mathcal{C}^\infty$.

La construction du modèle RBF à gradients consiste alors à déterminer les coefficients w_{ij} de sorte que l'approximation soit interpolante non seulement pour les réponses mais également pour les gradients. Ainsi, le problème à résoudre s'écrit :

$$\begin{aligned} &\forall (k,l) \in [\![1,n_s]\!]\times[\![1,n_p]\!], \forall \mathbf{x}^{(k)} \in \mathcal{D}, \\ &\tilde{y}\big(\mathbf{x}^{(k)}\big) = \tilde{y}_k = y_k = y\big(\mathbf{x}^{(k)}\big) \qquad (3.96) \\ &\frac{\partial \tilde{y}}{\partial x_l}\big(\mathbf{x}^{(k)}\big) = \tilde{y}_{k,l} = y_{k,l} = \frac{\partial y}{\partial x_l}\big(\mathbf{x}^{(k)}\big) \qquad (3.97) \end{aligned}$$

Les équations 3.96 et 3.97 conduisent à :

$$\sum_{i=1}^{n_s} w_i \phi_{ki} + \sum_{j=1}^{n_p}\sum_{i=1}^{n_s} w_{ij}\phi_{ki,j} = y_k \qquad (3.98)$$

$$\sum_{i=1}^{n_s} w_i \phi_{ki,l} + \sum_{j=1}^{n_p}\sum_{i=1}^{n_s} w_{ij}\phi_{ki,jl} = y_{k,l} \qquad (3.99)$$

Ce qui conduit à la formulation matricielle suivante :

$$\mathbf{\Phi}_g \mathbf{w}_g = \mathbf{y}_g \qquad (3.100)$$

avec $\mathbf{\Phi}_g = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Phi}_d \\ -\mathbf{\Phi}_d^T & \mathbf{\Phi}_{dd} \end{bmatrix}$ où l'on retrouve la matrice $\mathbf{\Phi}$ apparaissant dans le cadre des RBF sans gradient et les matrices $\mathbf{\Phi}_d$ et $\mathbf{\Phi}_{dd}$ faisant intervenir les dérivées première et secondes des fonctions de base radiales. On constate ici que la matrice $\mathbf{\Phi}_g$ n'est plus symétrique et que sa définie positivité n'est plus assurée (rendant le calcul de la variance

impossible).

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n_s} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n_s} \\ \vdots & & \ddots & \vdots \\ \phi_{n_s1} & \phi_{n_s2} & \cdots & \phi_{n_sn_s} \end{bmatrix}$$

$$\boldsymbol{\Phi}_d = \begin{bmatrix} \phi_{11,1} & \phi_{11,2} & \cdots & \phi_{11,n_p} & \phi_{12,1} & \cdots & \phi_{1n_s,n_p} \\ \phi_{21,1} & \phi_{21,2} & \cdots & \phi_{21,n_p} & \phi_{22,1} & \cdots & \phi_{2n_s,n_p} \\ \vdots & & \ddots & \vdots & & \ddots & \\ \phi_{n_s1,1} & \phi_{n_s1,2} & \cdots & \phi_{n_s1,n_p} & \phi_{n_s2,1} & \cdots & \phi_{n_sn_s,n_p} \end{bmatrix}$$

$$\boldsymbol{\Phi}_{dd} = \begin{bmatrix} \phi_{11,11} & \phi_{11,12} & \cdots & \phi_{11,1n_p} & \phi_{12,11} & \cdots & \phi_{1n_s,1n_p} \\ \phi_{11,21} & \phi_{11,22} & \cdots & \phi_{11,2n_p} & \phi_{12,21} & \cdots & \phi_{1n_s,2n_p} \\ \vdots & & \ddots & \vdots & & \ddots & \\ \phi_{11,n_p1} & \phi_{11,n_p2} & \cdots & \phi_{11,n_pn_p} & \phi_{12,n_p1} & \cdots & \phi_{1n_s,n_pn_p} \\ \phi_{21,11} & \phi_{21,12} & \cdots & \phi_{21,1n_p} & \phi_{22,11} & \cdots & \phi_{2n_s,1n_p} \\ \vdots & & \ddots & \vdots & & \ddots & \\ \phi_{n_s1,n_p1} & \phi_{n_s1,n_p2} & \cdots & \phi_{n_s1,n_pn_p} & \phi_{n_s2,n_p1} & \cdots & \phi_{n_sn_s,n_pn_p} \end{bmatrix}$$

$$\mathbf{w}_g = \begin{bmatrix} w_1 & \cdots & w_{n_s} & w_{11} & w_{12} & \cdots & w_{1n_p} & w_{21} & \cdots & w_{n_sn_p} \end{bmatrix}^T$$

$$\mathbf{y}_g = \begin{bmatrix} y_1 & \cdots & y_{n_s} & y_{11} & y_{12} & \cdots & y_{1n_p} & y_{21} & \cdots & y_{n_sn_p} \end{bmatrix}^T$$

■ **Détermination des paramètres :** Comme dans le cadre des RBF sans gradient, le problème comporte n_p paramètres internes à obtenir. Ceux-ci, assurant la définition de la fonction gaussienne spécifiée par l'équation 3.62 (page 159), sont déterminés en employant une approche de type validation croisée (cf. partie 2.2.3 page 159).

3.3 Autres métamodèles à gradients

Parmi les autres modèles de substitution capables de prendre en compte les gradients de manière directe, on trouve : des versions à gradients des SVR (*Suport Vector Regression*) [LÁZARO et al. 2005 ; JAYADEVA, KHEMCHANDANI et CHANDRA 2006 ; BOMPARD 2011], des fonctions de Shepard [LEARY, BHASKAR et KEANE 2004a] et des approximations de type MLS (*Moving Least Square*) [C. KIM, S. WANG et K. K. CHOI 2005] et WLS (*Weighting Least Square*) [van KEULEN et VERVENNE 2004].

4 Comparaisons de métamodèles

Dans cette partie, on s'intéresse aux performances en termes de qualité d'approximation des différents métamodèles introduits précédemment. Pour cela, on fera ici usage uniquement de fonctions analytiques que l'on cherche à approcher avec des métamodèles de type krigeage et RBF avec et sans gradients. Dans un premier temps et afin de fixer

visuellement les performances que peuvent fournir ces modèles de substitution, on s'intéressera à des exemples de fonctions à une et deux variables. On s'intéressera par la suite aux performances d'approximation pour des fonctions comportant un nombre de variables plus important.

Les abréviations regroupées dans le tableau 3.6 seront utilisées dans cette partie et permettront d'alléger les notations.

Métamodèle	Abréviation
Fonctions de base radiales	RBF
Fonctions de base radiales à gradients	GRBF
Fonctions de base radiales à gradients indirectes	InRBF
Krigeage ordinaire	KO
Cokrigeage ordinaire à gradients	CKO
Cokrigeage ordinaire à gradients indirect	InCKO

TABLE 3.6 – Abréviations employées

4.1 Fonction d'une variable

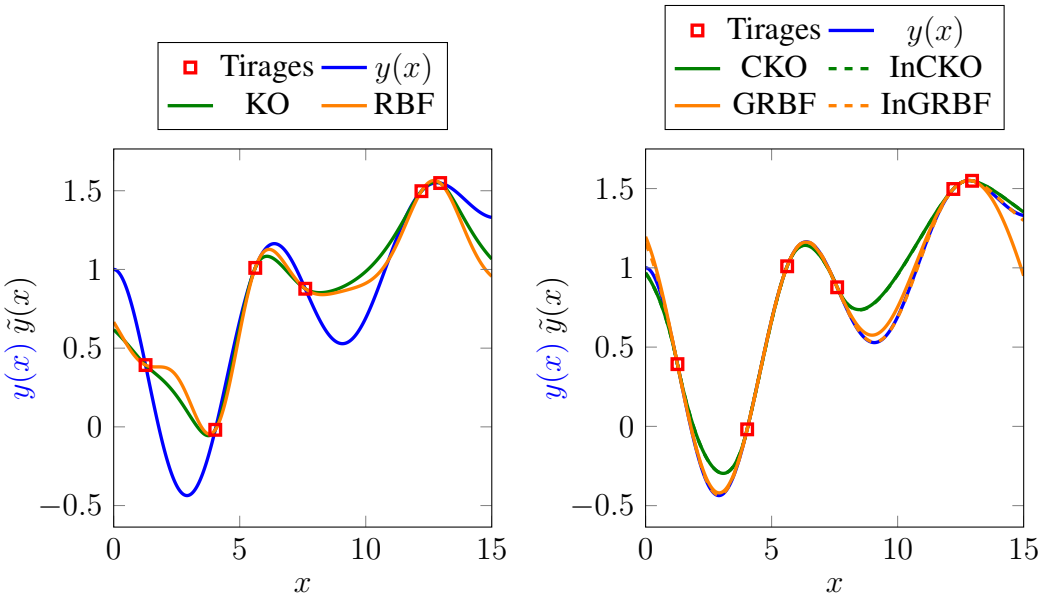
Dans l'objectif d'illustrer visuellement les approximations fournies par les différentes méthodes, on considère ici le même exemple que celui introduit dans la partie 2.1.6 page 147 soit la fonction à une variable $y(x) = \exp(-x/10)\cos(x) + x/10$.

On distinguera d'un coté les métamodèles sans gradient de type krigeage ordinaire et RBF et de l'autre ceux les prenant en compte de type cokrigeage ordinaire, RBF à gradients et les approches indirectes (krigeage et RBF). On considérera, pour la construction de cette approximation, que les métamodèles de type krigeage et RBF sont construits en utilisant respectivement la fonction de corrélation Matérn ($\nu = 3/2$) et la fonction gaussienne.

4.1.1 Réponses

En considérant un tirage LHS comportant 6 points, les 6 modèles de substitution sont construits pour approcher la fonction à une variable considérée. Les approximations obtenues sont regroupées dans la figure 3.10. Comme on peut l'observer à partir de cet exemple, la prise en compte des gradients permet, comme attendu, d'obtenir une approximation de qualité nettement meilleure. Comme le confirme le tableau 3.7, les métamodèles de type RBF fournissent ici une approximation de meilleure qualité que les métamodèles basés sur le krigeage. Dans ce cas particulier, on observe également que le modèle de RBF à gradients indirect parvient à reconstruire de manière quasi-exacte la fonction initiale.

Par ailleurs, on constate que pour le cokrigeage, les approches directe et indirecte fournissent des approximations de qualité très similaire. Dans le cas des RBF, les qualités sont proches mais un écart subsiste.



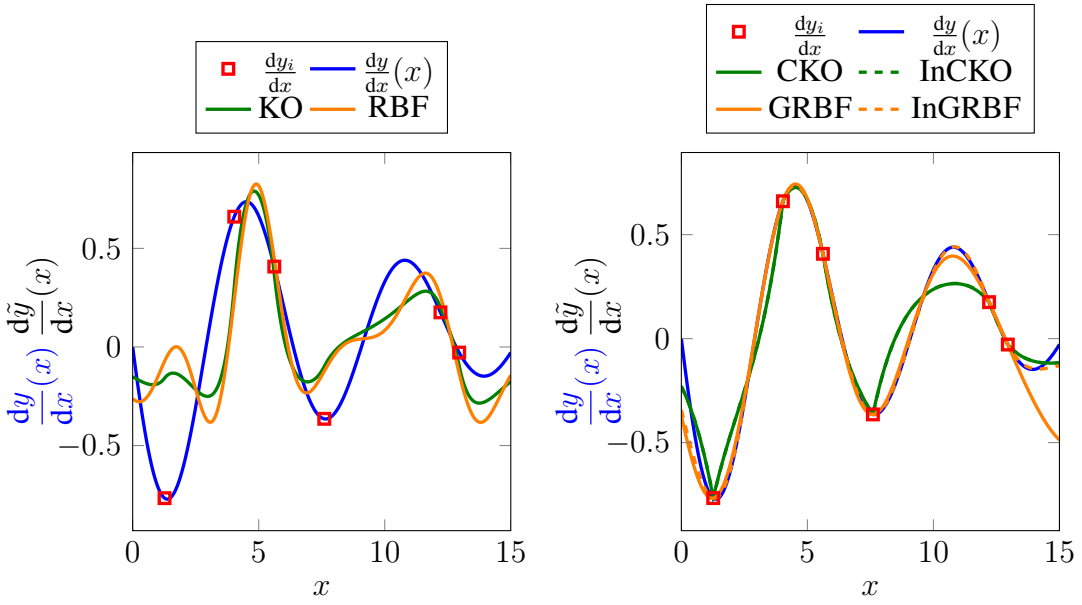
(a) Métamodèles sans gradient **(b)** Métamodèles à gradients

FIGURE 3.10 – Approximations avec et sans gradients d'une fonction d'une variable (échantillonnage LHS, $n_s = 6$)

	RMSE	R^2	RMAE	RAAE	Q_3
KO	0.2376	0.8646	1.2491	0.3822	$2.35 \cdot 10^{-2}$
RBF	0.2612	0.8344	1.6081	0.4298	$2.84 \cdot 10^{-2}$
CKO	0.1084	0.9758	0.5144	0.1234	$4.9 \cdot 10^{-3}$
GRBF	$6.72 \cdot 10^{-2}$	0.987	0.6709	$5.77 \cdot 10^{-2}$	$1.9 \cdot 10^{-3}$
InCKO	0.1076	0.9762	0.51	0.1225	$4.8 \cdot 10^{-3}$
InGRBF	$\mathbf{1.86 \cdot 10^{-2}}$	**0.999**	**0.2642**	$\mathbf{1.05 \cdot 10^{-2}}$	$\mathbf{1 \cdot 10^{-4}}$

TABLE 3.7 – Critères de qualité globaux des approximations de la figure 3.10 (la signification des critères peut être trouvée page 85)

La figure 3.11 présente, quant à elle, les dérivées fournies par les métamodèles. Cette figure confirme le résultat issu de la mise en œuvre des modèles à gradients directs qui sont interpolants non seulement pour les réponses mais également pour les dérivées.



(a) Métamodèles sans gradient **(b)** Métamodèles à gradients

FIGURE 3.11 – Dérivées des approximations (échantillonnage LHS, $n_s = 6$)

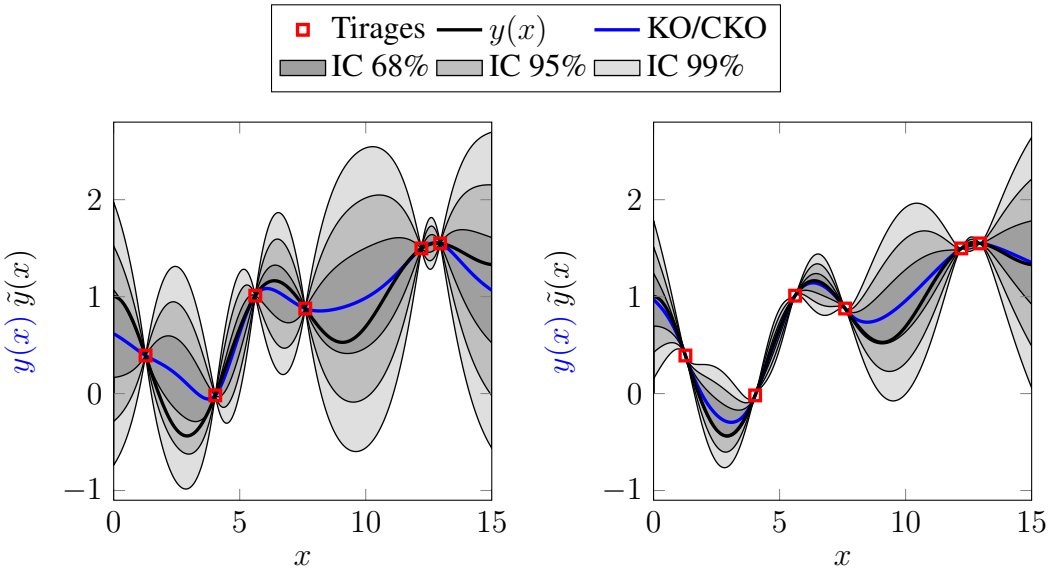
4.1.2 Intervalles de confiance

Grâce aux formulations employées et comme nous l'avons introduit précédemment, il est possible d'élaborer des intervalles de confiance employant l'écart type de l'approximation fourni par chacune des méthodes. On notera cependant l'impossibilité du modèle GRBF à fournir cette information en raison des mauvaises propriétés de la matrice des fonctions de base (il ne sera donc pas pris en compte dans cette partie). Les résultats obtenus sont présentés dans les figures 3.12 et 3.13 pour les trois intervalles de confiance introduits dans la partie 2.1.3 page 143.

On constate à l'observation de ces courbes :

- que pour les approches sans gradient, le krigeage présente une approximation plus faible de l'incertitude.
- que la prise en compte des gradients dans l'approche de krigeage assure une réduction de l'incertitude. Ce constat est conforme aux attentes dans le sens où l'on apporte des informations complémentaires lors de la prise en compte des gradients.

Remarque 3.16 : Ces notions d'incertitude sont bien sûr à considérer comme une information qualitative. En effet, elles fournissent une information statistique pour des modèles construits à l'aide d'informations déterministes.



(a) Krigeage **(b)** Cokrigeage à gradients

FIGURE 3.12 – Approximations de type krigeage et intervalles de confiance (échantillonnage LHS, $n_s = 6$)

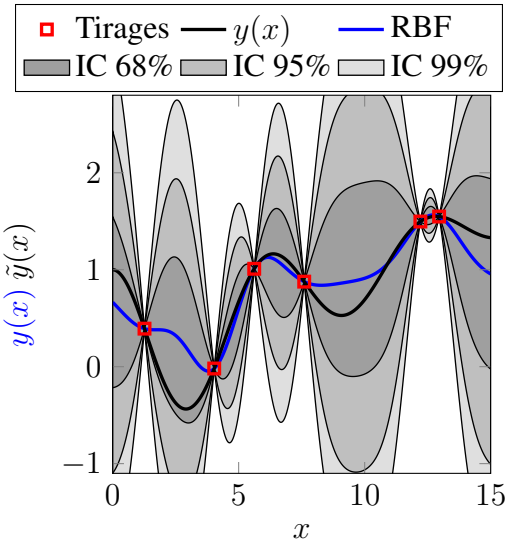


FIGURE 3.13 – Approximations de type RBF et intervalles de confiance (échantillonnage LHS, $n_s = 6$)

4.2 Fonction de deux variables

Lors du passage à deux variables, la complexité des modèles de substitution augmente. Plus précisément, il est alors possible de considérer un paramètre de modèle pour chacune des n_p variables. Dans un premier temps, nous nous intéresserons à la construction de modèles de substitution que l'on caractérisera d'*isotropes* : on considérera le même paramètre (longueur de corrélation) pour chacune des variables. On montrera par la suite qu'une fonction présentant de grosses dissymétries par rapport aux deux variables nécessite la mise en œuvre de modèles de substitution *anisotropes*.

Pour mener à bien cette étude, quatre fonctions test analytiques seront considérées : présentées en figure 3.14, il s'agit des fonctions *Rosenbrock* [5], *Branin* [6], *Six-hump Camel Back* [7] et *Schwefel* [8]. Elles présentent des difficultés différentes en termes de non-linéarité, de nombre de minima globaux et locaux. Quelques caractéristiques de ces fonctions sont données dans le tableau 3.8.

Nom	n_p	$\mathcal{D}$	Min. locaux	Min. globaux
Branin	2	$[-5, 10] \times [0, 15]$	0	3
Six-hump Camel Back	2	$[-2, 2] \times [-1, 1]$	4	2
Rosenbrock	n_p	$[-2, 2]^{n_p}$	Peu[a]	1
Schwefel	n_p	$[-500, 500]^{n_p}$	Nombreux	1

[a] Dépend du nombre de variables.

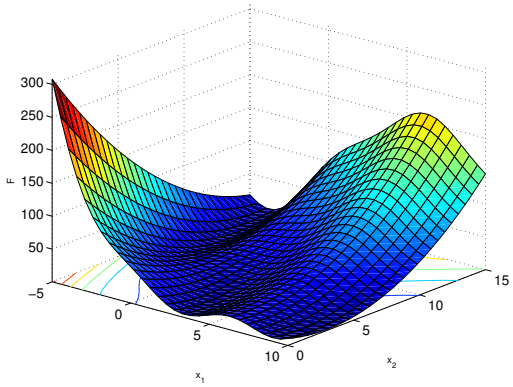
TABLE 3.8 – Fonctions test employées pour évaluer la qualité d'approximation des modèles de substitution

Remarque 3.17 : Il est important de remarquer que certaines de ces fonctions ne sont pas représentatives de celles que l'on rencontre classiquement pour un problème mécanique (en particulier la fonction *Schwefel*), mais permettent d'illustrer les performances en termes de qualité d'approximation.

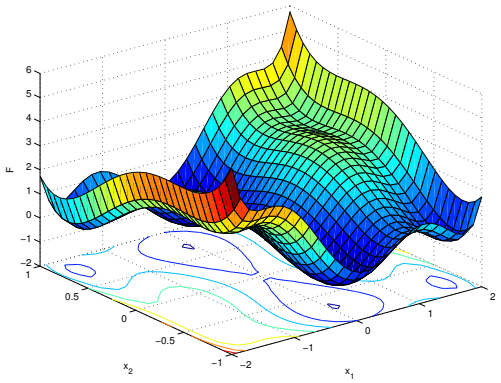
4.2.1 Prise en compte de l'anisotropie

Considérons ici les fonctions *Branin* et *Six-Hump Camel Back* pour illustrer la nécessité de prendre en compte l'anisotropie dans le modèle de substitution. Les approximations construites ici ne seront que des approches sans gradient (le comportement anisotrope sera

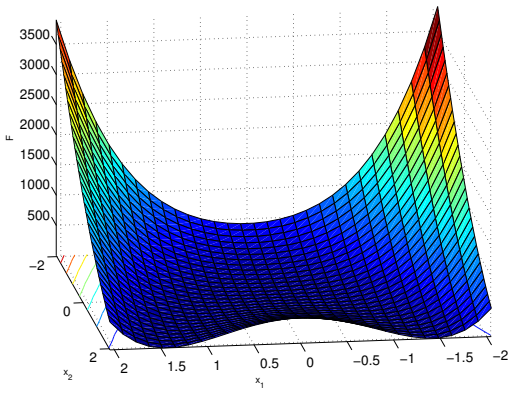
5. $\forall \mathbf{x} \in [-2, 2]^{n_p}$, $y(\mathbf{x}) = \sum_{i=1}^{n_p - 1} \left[100 \left(x_i^2 - x_{i+1} \right)^2 + (x_i - 1)^2 \right]$
6. $\forall (x_1, x_2) \in [-5, 10] \times [0, 15]$, $y(x_1, x_2) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos(x_1) + 1$
7. $\forall (x_1, x_2) \in [-2, 2] \times [-1, 1]$, $y(x_1, x_2) = \left(4 - 2.1 x_1^2 + \frac{x_1^4}{3} \right) x_1^2 + x_1 x_2 + \left(-4 + 4 x_2^2 \right) x_2^2$
8. $\forall \mathbf{x} \in [-500, 500]^{n_p}$, $y(\mathbf{x}) = 418.9829 n_p - \sum_{i=1}^{n_p} \left[x_i \sin \left(\sqrt{|x_i|} \right) \right]$



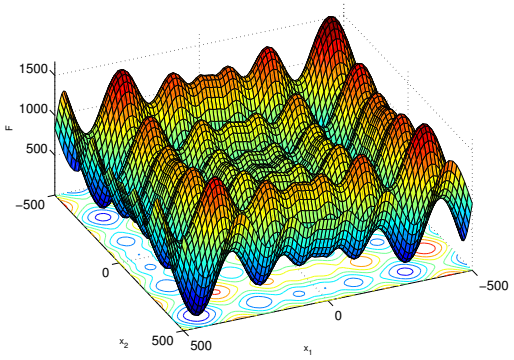
(a) *Branin*



(b) *Six-hump Camel Back*



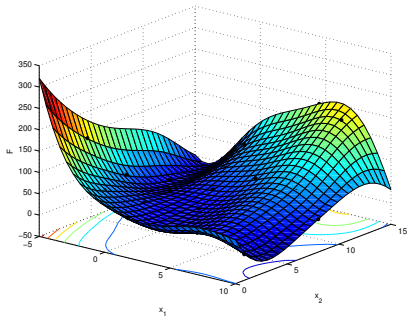
(c) *Rosenbrock*



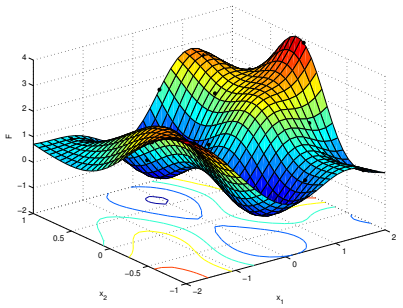
(d) *Schwefel*

FIGURE 3.14 – Fonctions test

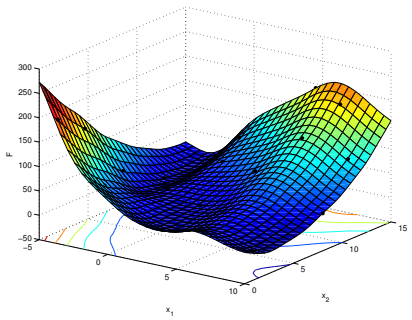
bien extensible aux modèles de substitution avec gradients) de type krigeage ordinaire et RBF. Un tirage IHS (cf. page 70) comportant 30 points est réalisé pour construire des métamodèles de krigeage ordinaire et RBF. La figure 3.15 présente les approximations obtenues avec un seul paramètre. Elle est à comparer avec les modèles anisotropes présentés dans la figure 3.16.



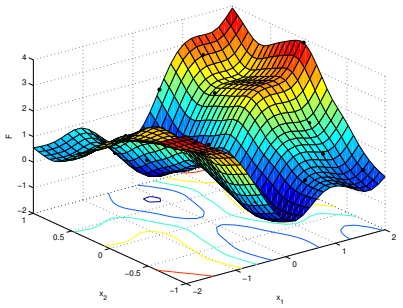
(a) *Branin* (RBF)



(b) *Six-hump Camel Back* (RBF)



(c) *Branin* (KO)



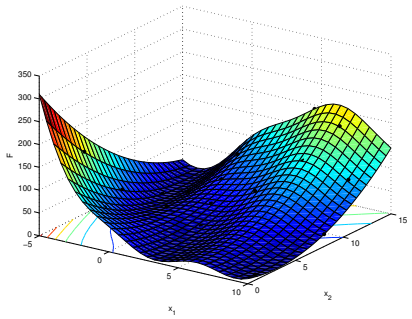
(d) *Six-hump Camel Back* (KO)

FIGURE 3.15 – Métamodèles isotropes

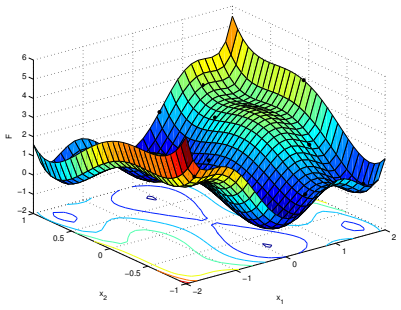
On observe, en comparant ces deux figures, que la prise en compte de l'anisotropie permet d'accroître la qualité d'approximation des métamodèles. Ce constat est frappant pour les approximations anisotropes de la fonction *Six-Hump Camel Back* qui s'approchent grandement de la fonction réelle (voir et comparer par exemple les figures 3.14b, 3.15b et 3.16b). Le tableau 3.9 confirme ce constat.

Le tableau donne, par ailleurs, les valeurs des paramètres internes l des métamodèles pour les deux fonctions dans les cas isotropes et anisotropes. On constate que cette valeur peut différer nettement entre les approches isotropes et anisotropes.

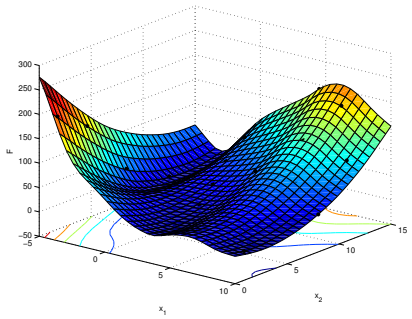
Suite à cette étude et dans le reste des travaux présentés dans ce manuscrit, il sera toujours fait usage de modèles de substitution anisotropes intégrant autant de paramètres que le problème comporte de variables (n_p).



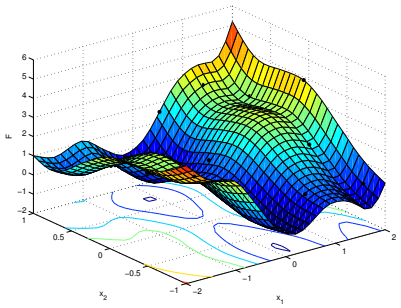
(a) *Branin* (RBF)



(b) *Six-hump Camel Back* (RBF)



(c) *Branin* (KO)



(d) *Six-hump Camel Back* (KO)

FIGURE 3.16 – Métamodèles anisotropes

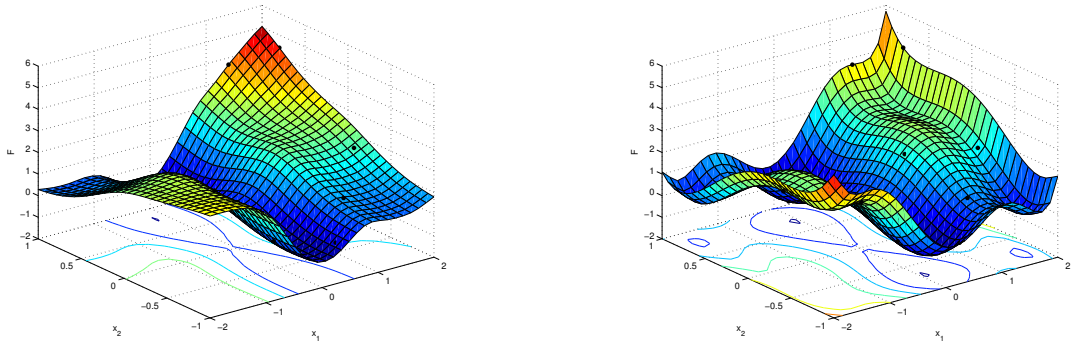
		RBF (iso)	RBF (aniso)	KO (iso)	KO (aniso)
Branin	l	0.254	$(0.253, 2.30 \cdot 10^{-4})$	0.198	$(0.283, 1.244)$
	RMSE	0.120	0.480	4.71	5.55
	R^2	0.952	0.9999	0.993	0.990
	RAAE	0.106	$2.88 \cdot 10^{-3}$	$5.00 \cdot 10^{-2}$	$4.71 \cdot 10^{-2}$
	RMAE	2.12	$7.39 \cdot 10^{-2}$	0.659	0.726
	Q_3	$1.51 \cdot 10^{-3}$	$2.43 \cdot 10^{-6}$	$2.34 \cdot 10^{-4}$	$3.25 \cdot 10^{-4}$
Six-Hump C. Back	l	0.973	$(0.011, 3.9 \cdot 10{-}3)$	0.348	$(0.144, 0.071)$
	RMSE	0.561	$2.25 \cdot 10^{-2}$	0.362	0.272
	R^2	0.811	**0.9997**	0.923	**0.956**
	RAAE	0.273	$8.79 \cdot 10^{-3}$	0.181	0.124
	RMAE	3.81	0.148	1.68	1.28
	Q_3	$9.57 \cdot 10^{-3}$	$\mathbf{1.53 \cdot 10^{-5}}$	$3.99 \cdot 10^{-3}$	$\mathbf{2.25 \cdot 10^{-3}}$

TABLE 3.9 – Caractéristiques des métamodèles isotropes et anisotropes

4.2.2 Prise en compte des gradients

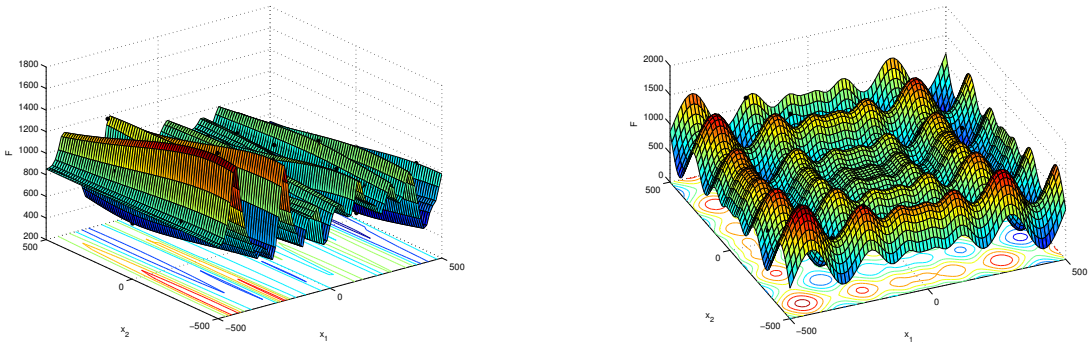
Considérons désormais les fonctions *Six-Hump Camel Back* et *Schwefel*, que l'on cherchera à approximer à l'aide de métamodèles avec et sans gradients en utilisant des tirages comportant respectivement 20 et 40 points (tirage IHS). On emploiera dans cette partie des métamodèles de types krigeage et RBF sans gradient et du même type avec gradients par approche directe et indirecte.

Afin de ne pas surcharger ce document, seules les surfaces de réponses de type krigeage et cokrigeage direct sont proposées dans les figures 3.17 et 3.18. Les performances en termes de qualité d'approximation de tous les métamodèles sont présentées dans le tableau 3.10.



(a) KO **(b)** CKO

FIGURE 3.17 – Fonction *Six-Hump Camel Back* approchée par métamodèles de krigeage avec et sans gradients (échantillonnage IHS, $n_s = 20$)



(a) KO **(b)** CKO

FIGURE 3.18 – Fonction *Schwefel* approchée par métamodèles de krigeage avec et sans gradients (échantillonnage IHS, $n_s = 40$)

		RBF	GRBF	InRBF	KO	CKO	InKO
S.H.C.B.	RMSE	0.581	7.05	0.712	0.471	0.151	0.147
	R^2	0.796	$4.63 \cdot 10^{-3}$	0.770	0.868	**0.987**	0.988
	RAAE	0.3251	0.458	0.125	0.252	$3.86 \cdot 10^{-2}$	$3.68 \cdot 10^{-2}$
	RMAE	3.04	8.85	5.48	2.55	0.902	0.869
	$\mathtt{Q}_3$	$1.03{\cdot}10^{-2}$	1.51	$1.54 \cdot 10^{-2}$	$6.74 \cdot 10^{-3}$	$\mathbf{6.92 \cdot 10^{-4}}$	$6.56 \cdot 10^{-4}$
Schwefel	RMSE	275	2104	191	238	10.2	22.1
	R^2	0.140	0.135	0.586	0.294	**0.999**	0.994
	RAAE	1.06	0.702	0.466	0.936	$2.94 \cdot 10^{-2}$	$5.95 \cdot 10^{-2}$
	RMAE	3.85	3.55	2.70	3.37	0.1761	0.506
	$\mathtt{Q}_3$	$2.78 \cdot 10^{-2}$	1.63	$1.34 \cdot 10^{-2}$	$2.08 \cdot 10^{-2}$	$\mathbf{3.83 \cdot 10^{-5}}$	$1.79 \cdot 10^{-4}$

TABLE 3.10 – Caractéristiques des métamodèles avec et sans gradients en 2D sur les fonctions test *Six-Hump Camel Back* (S.H.C.B.) et *Schwefel* construits avec des tirages IHS comportant respectivement 20 et 40 points

On constate ici que l'ajout des gradients a des effets variables en fonction des deux métamodèles :

- dans le cas du krigeage, l'apport des gradients est indiscutable : il permet d'obtenir une prédiction de qualité nettement meilleure. Ce constat est par ailleurs particulièrement marqué sur la fonction *Schwefel* qui s'avère être une fonction complexe à approximer. Le cokrigeage direct et le cokrigeage indirect permettent une approximation de qualité d'extrêmement bonne avec 40 réponses et 40 gradients (les gradients ont chacun deux composantes). La fonction *Schwefel* s'avère en fait particulièrement adaptée au krigeage en raison de son principe de construction consistant à procéder à des corrections autour d'une valeur moyenne (constante car on emploie un modèle de krigeage et cokrigeage ordinaire). Dans le cas de la fonction *Six-Hump Camel Back*, les constats concernant le krigeage sont les mêmes.

- dans le cas des RBF, la prise en compte des gradients s'avère problématique et le métamodèle ne parvient pas à fournir une approximation de meilleure qualité. Le constat est d'autant plus mauvais que le métamodèle RBF à gradients dégrade l'approximation lors de la prise en compte des gradients. Un nouvel exemple d'approximation avec le modèle RBF, mais en considérant un nombre de points plus important, montre que le métamodèle peut parvenir à fournir une approximation de qualité (cf. tableau 3.11 pour une approximation de la fonction *Six-Hump Camel Back* en utilisant 30 points échantillonnés). L'apport des gradients dans ce cas reste néanmoins discutable.

La version indirecte de l'approximation à gradients permet, dans le cas du cokrigeage, d'obtenir une approximation de qualité équivalente à celle du cokrigeage direct sur les deux fonctions. Dans le cas des RBF, sur la fonction *Schwefel*, l'approximation par approche indirecte est légèrement meilleure que l'approche directe, mais pose toujours problème dans le cas de la fonction *Six-Hump Camel Back*.

		RBF	GRBF	InRBF	KO	CKO	InKO
S.H.C.B.	RMSE	$3.13 \cdot 10^{-2}$	0.173	0.400	0.272	$7.83 \cdot 10^{-2}$	$7.41 \cdot 10^{-2}$
	R^2	0.9994	0.982	0.906	0.956	**0.997**	**0.997**
	RAAE	$9.89 \cdot 10^{-3}$	$6.89 \cdot 10^{-2}$	$7.84 \cdot 10^{-2}$	0.124	$1.60 \cdot 10^{-2}$	$1.86 \cdot 10^{-2}$
	RMAE	0.192	0.744	3.68	1.29	0.421	0.400
	$\mathtt{Q}_3$	$2.99 \cdot 10^{-5}$	$9.06 \cdot 10^{-4}$	$4.87 \cdot 10^{-3}$	$2.25 \cdot 10^{-3}$	$\mathbf{1.87 \cdot 10^{-4}}$	$\mathbf{1.67 \cdot 10^{-4}}$

TABLE 3.11 – Caractéristiques des métamodèles avec et sans gradients en 2D sur la fonction test *Six-Hump Camel Back* construits avec un tirage IHS comportant respectivement 30 points (en comparaison des résultats obtenus avec 20 points dans le tableau 3.10)

■ **Bilan :** Il apparaît, sur les fonctions présentées ici, que le cokrigeage permet de fournir une approximation dans tous les cas de figure. Plus précisément, il parvient à approcher la fonction *Schwefel*. Les métamodèles RBF à gradients ne se comportent pas de manière optimale sur ces exemples : ils nécessitent un nombre de points plus important que le krigeage pour fournir une approximation de qualité.

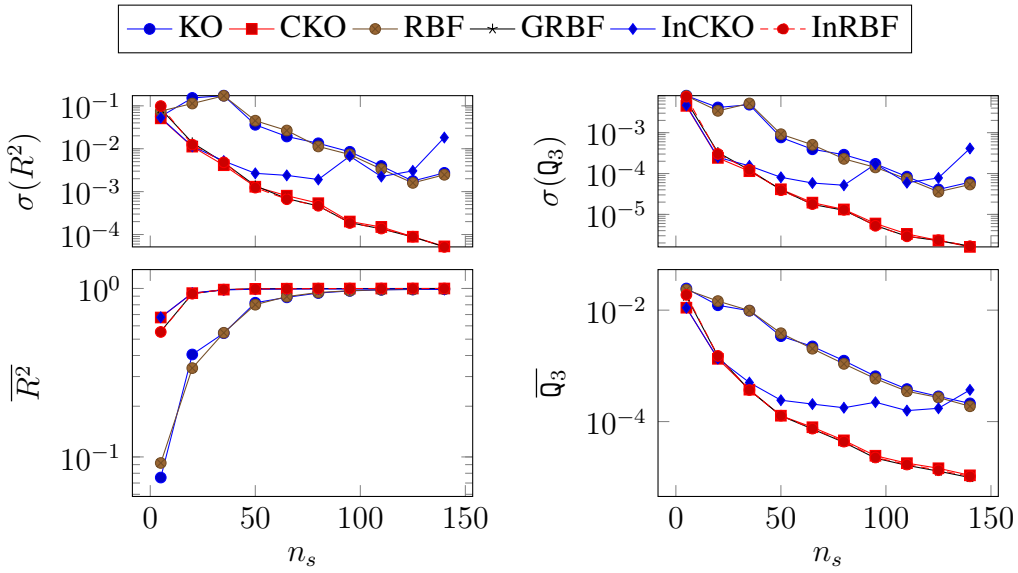
4.3 Dimensions supérieures

Afin d'étudier les performances d'approximation des modèles de substitution pour des nombres de variables plus importants, on considère ici les fonctions *Rosenbrock* et *Schwefel* en dimensions 5 et 10 dont les expressions sont données dans le tableau 3.8. L'étude proposée se base sur l'analyse de la qualité d'approximation fournie par des métamodèles de types RBF et du krigeage avec et sans gradients. Pour cela, on considère des tirages IHS comportant entre 5 et 200 points. Chaque tirage est exécuté 50 fois et pour chacun des tirages obtenus, les métamodèles sont construits. Leur qualité est ensuite évaluée en calculant moyennes et écart-types des critères R^2 et $\mathtt{Q}_3$ sur chacune des 50 exécutions. Afin de ne par surcharger les résultats, seules les fonctions test ayant 5 et 10 variables seront prises en compte. Les résultats obtenus sont présentés dans les figures 3.19, 3.20, 3.21 et 3.22.

À partir de ces courbes, il est possible de faire de nombreux constats :

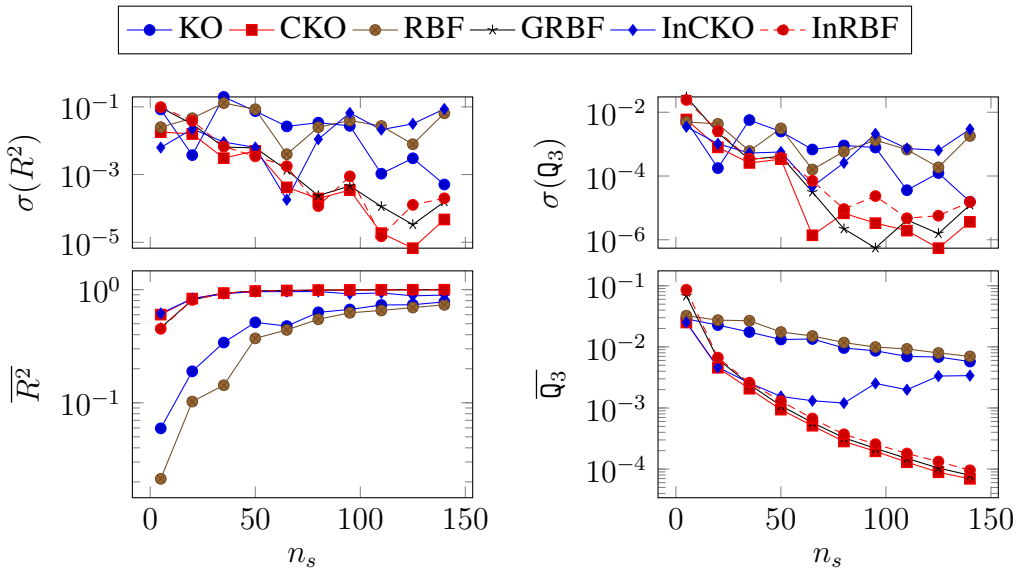
- dans le cas de la fonction de *Rosenbrock*, qui est une fonction peu complexe à approximer de manière globale (localement cela peut s'avérer plus complexe), les approximations parviennent à fournir une approximation de qualité de la fonction avec ou sans gradients et en dimension 5 ou 10.

- dans le cas de la fonction *Schwefel*, qui est une fonction extrêmement complexe à approximer, seul le cokrigeage direct parvient à donner une approximation correcte de la fonction. Cela confirme le résultat obtenu en dimension 2. Par ailleurs, le métamodèle de type RBF à gradients est mis en défaut à de nombreuses reprises en dimension 10.



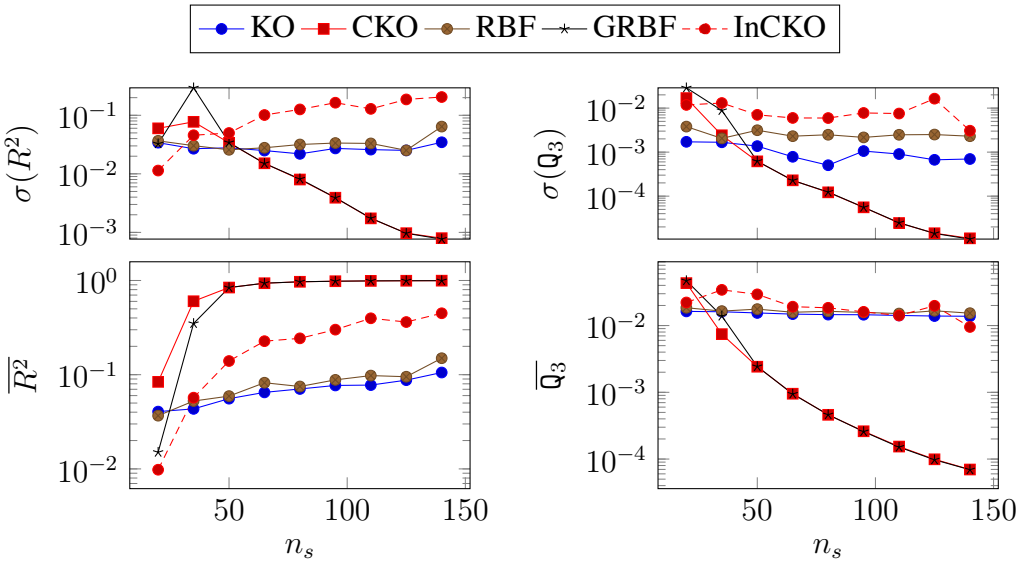
(a) Moyennes et écart-types du critère R^2 **(b)** Moyennes et écart-types du critère $\mathtt{Q}_3$

FIGURE 3.19 – Qualité d'approximation de la fonction Rosenbrock à 5 variables obtenue avec 50 exécutions de chaque tirage



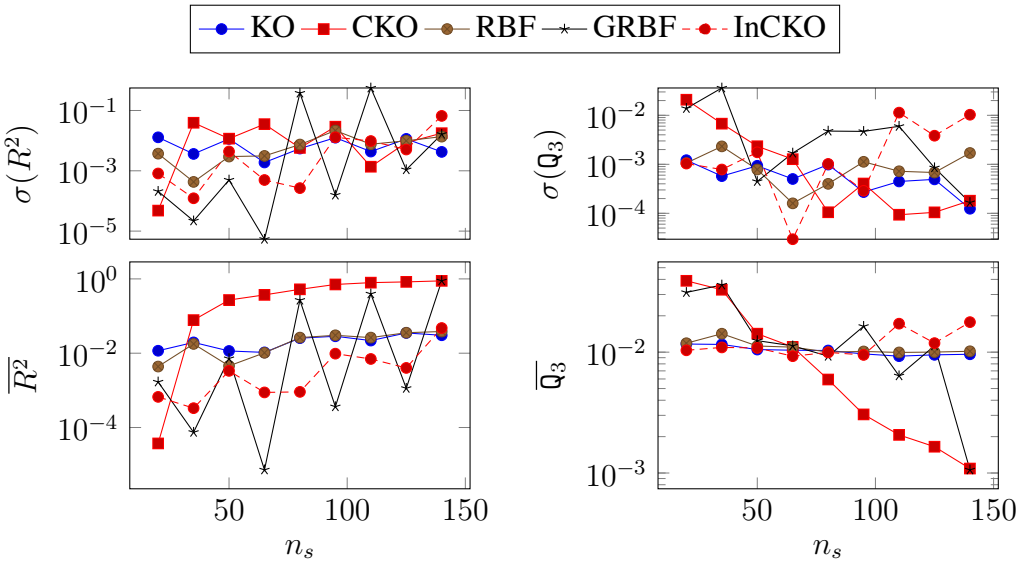
(a) Moyennes et écart-types du critère R^2 **(b)** Moyennes et écart-types du critère $\mathtt{Q}_3$

FIGURE 3.20 – Qualité d'approximation de la fonction Rosenbrock à 10 variables obtenue avec 50 exécutions de chaque tirage



(a) Moyennes et écart-types du critère R^2 **(b)** Moyennes et écart-types du critère $\mathtt{Q}_3$

FIGURE 3.21 – Qualité d'approximation de la fonction Schwefel à 5 variables obtenue avec 50 exécutions de chaque tirage



(a) Moyennes et écart-types du critère R^2 **(b)** Moyennes et écart-types du critère $\mathtt{Q}_3$

FIGURE 3.22 – Qualité d'approximation de la fonction *Schwefel* à 10 variables obtenue avec 50 exécutions de chaque tirage

- les métamodèles à gradients indirects parviennent à fournir une approximation de qualité uniquement sur la fonction *Rosenbrock*. On distingue alors la version à gradients indirecte des RBF qui fournit une approximation équivalente aux métamodèles à gradients directs sur cette fonction et pour les deux dimensions, de la version indirecte du cokrigeage qui ne fournit une approximation équivalente à l'approche directe que pour un nombre de tirages faible pour les deux dimensions (jusqu'à $n_s = 50$ environ). Ce phénomène dû à la dégradation du conditionnement de la matrice de covariance, ne s'observe que sur le critère $\mathtt{Q}_3$.

- en l'absence de gradient, les métamodèles nécessitent un nombre très important de points échantillonnés pour parvenir à construire une approximation de la fonction *Schwefel*.

Du point de vue de la robustesse, le cokrigeage direct présente globalement une meilleure capacité approximation tout en étant moins sensible au caractère aléatoire du tirage. En effet, sur l'ensemble des cas-tests, il minimise l'écart-type pour chaque valeur de n_s. Si on ne considère que des exemples plus proches de ceux rencontrés dans le cadre d'une optimisation mécanique, l'approche GRBF permet d'obtenir une approximation comparable au cokrigeage.

Par ailleurs, dans le cas d'une fonction complexe à approcher, les métamodèles à gradients peuvent fournir une approximation de plus mauvaise qualité que les approches sans gradient. Il est donc important de noter que, comme pour toute approximation par métamodèle, une attention particulière doit être accordée sur le choix du nombre d'échantillons.

4.4 Bilan

Au vu de ces très nombreux résultats, on s'aperçoit de la qualité d'approximation que peuvent fournir les métamodèles pour différentes dimensions et des complexités de fonctions à approcher extrêmement variables. En particulier dans le cas des métamodèles à gradients, le temps de construction de ces métamodèles peut devenir extrêmement important en raison, des multiples inversions de matrices de taille importante qui doivent être faites pour trouver le jeu de paramètres internes optimal.

On peut alors conclure par plusieurs points :

- l'incorporation des gradients dans les modèles de substitution peut permettre de construire des approximations de très bonne qualité voire même d'approximer des fonctions que les approches sans gradients ne parviennent pas à approcher.

- la prise en compte des gradients de manière directe semble la plus appropriée pour des raisons numériques et de coût de construction. Elles présentent par ailleurs une meilleure robustesse.

- pour des dimensions importantes et pour une fonction complexe, le cokrigeage semble le plus adapté.

Au-delà de ces considérations de qualité de prédiction, on privilégiera l'utilisation de modèles de substitution de type krigeage en raison de la plus grande robustesse dans la détermination des paramètres. En effet, la très forte multimodalité de l'erreur de validation croisée à minimiser dans le cadre des RBF engendre un coût de construction extrêmement important dû à l'usage d'un algorithme de minimisation globale.

Remarque 3.18 : Il n'existe pas de toolbox vraiment complète permettant la construction de modèle de substitution à gradients. On trouve cependant quelques éléments de code MATLAB en lien avec l'ouvrage [FORRESTER, SÓBESTER et KEANE 2008]. C'est pour cette raison que la toolbox GRENAT (***GR**adient **EN**hanced **A**pproximation **T**oolbox*) a été développée lors de cette thèse. Adaptée à MATLAB, elle intègre les métamodèles de types krigeage, cokrigeage, RBF et GRBF introduits dans ce chapitre. Elle regroupe l'ensemble des routines nécessaires à la construction et à l'évaluation de ces métamodèles. De plus, elle intègre un certain nombre de dispositifs permettant d'obtenir de manière aisée et systématique les échantillonnages (en particulier, elle s'interface avec R pour obtenir certains tirages spécifiques) et de post-traiter les résultats. Par ailleurs, elle s'interface parfaitement avec la toolbox COMET qui assure le pilotage de COFAST3D. Fruit de 3 ans et demi de développements, elle compte environ 120 fichiers, 6600 lignes de codes et 2300 commentaires. À l'instar de la toolbox COMET et dans l'objectif de pouvoir réaliser des évolutions collaboratives et pérennes, elle s'appuie sur un dispositif de versionnement de type `git` comportant à l'heure actuelle environ 750 actions dans les sources (ajout, suppression de code, débogage...).

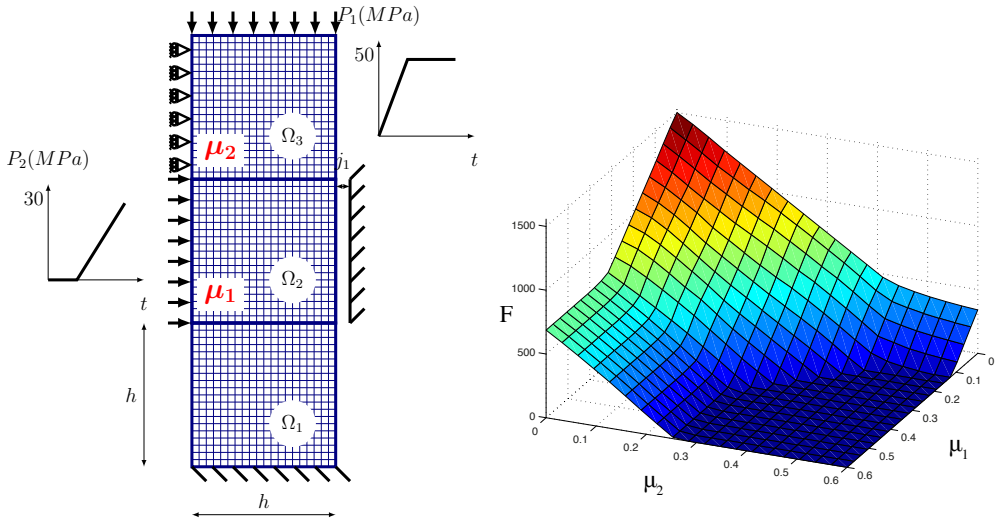
5 Métamodèles et stratégie multiparamétrique

On s'intéresse ici à l'emploi de la stratégie multiparamétrique pour construire l'approximation d'une fonction objectif issue de la résolution d'un problème d'assemblage. L'objectif consiste ici à étudier la relation entre la qualité d'approximation fournie par les modèles de substitution et le coût nécessaire pour construire ces derniers en termes de temps CPU.

5.1 Illustration sur un exemple simple

Dans un premier temps et pour faciliter la compréhension, considérons le cas-test CSMA introduit dans la partie 2.3.2 page 98 et dont la géométrie et la surface de réponse prises en compte sont rappelées dans la figure 3.23 page 189.

On rappelle que ce cas-test compte deux paramètres qui sont les deux coefficients de frottement aux interfaces $\Omega_1 - \Omega_2$ et $\Omega_2 - \Omega_3$. La fonction étudiée est la résultante de l'effort appliqué par la sous-structure Ω_2 sur la paroi.

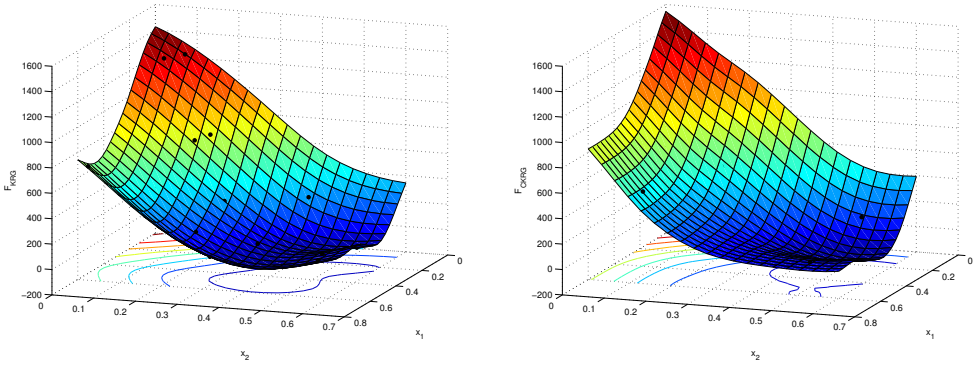


(a) Géométrie du cas-test CSMA **(b)** Résultant de l'effort à la paroi

FIGURE 3.23 – Rappels du cas-test CSMA

5.1.1 Exemple à nombre de tirages fixés

Dans un premier temps, considérons un nombre d'appels au solveur de 15. Le métamodèle de krigeage est construit à partir de 15 réponses fournies par le solveur COFAST3D tandis que le métamodèle de cokrigeage est construit à partir de 5 réponses et 5 gradients (nécessitant 10 appels au solveur pour être obtenus), soit un échantillonnage comportant 5 points. La figure 3.24 présente les surfaces de réponses obtenues.



(a) Krigeage ordinaire **(b)** Cokrigeage ordinaire

FIGURE 3.24 – Surfaces de réponse des modèles de substitution de type krigeage obtenues à partir de 15 appels au solveur

On constate visuellement les performances en termes de qualité de construction du cokrigeage pour approcher cette fonction. Ces résultats sont confirmés par les critères de qualité présentés dans le tableau 3.12b. On observe par ailleurs un accroissement du gain

de la stratégie multiparamétrique qui passe de 3.56 à 4.44 lors du calcul des gradients.

Gradients	non	oui
Nb. d'appels au solveur	15	15
Temps CPU moyen [s]	2.34s	1.89s
Temps CPU mini. [s]	0.17s	0.18s
Temps CPU maxi. [s]	8.01s	7.81s
Nb. itérations total	1636	1311
Nb. itérations moyen	109	87.4
Nb. itérations mini.s	5	5
Nb. itérations maxi.	381	371
Temps CPU total [s]	35.1s	28.4s
Temps CPU sans MPS [s]	125s	126s
Gain	3.56	4.44

(a) Caractéristiques des calculs mécaniques

	KO	CKO
R^2	0.9880	0.9781
RAAE	$8.884 \cdot 10^{-2}$	0.1120
RMAE	0.3774	0.4720
$\mathtt{Q}_1$	$8.917 \cdot 10^{-3}$	$1.395 \cdot 10^{-2}$
$\mathtt{Q}_2$	0.2990	0.5467
$\mathtt{Q}_3$	$7.475 \cdot 10^{-4}$	$1.367 \cdot 10^{-3}$

(b) Caractéristiques des métamodèles

TABLE 3.12 – Performances mécaniques et qualité d'approximation des métamodèles de krigeage et cokrigeage construits

La simplicité de la surface de réponse, qui est composée de plans, explique en grande partie la capacité du cokrigeage à fournir une approximation de qualité avec aussi peu de points.

5.1.2 Exemple à qualité fixée

Dans cette seconde étude, on s'intéresse à la construction d'une approximation de qualité similaire en considérant des tirages de type plan factoriel complet. Le métamodèle de krigeage est construit à partir d'un tirage comportant 9×9 points tandis que le cokrigeage est construit à partir de tirages comportant 3×3 et 5×5 points. Les caractéristiques des calculs mécaniques et les critères de qualité des métamodèles sont regroupés dans le tableau 3.13.

Dans ce cas, l'usage du cokrigeage permet d'obtenir une approximation de qualité équivalente au krigeage construit à partir de 9×9 en employant les réponses et gradients obtenus en 5×5 points. Il est important de noter que l'approximation obtenue par cokrigeage en 3×3 points possède une qualité très proche mais légèrement plus faible.

5.1.3 Montage fretté

Afin d'illustrer les capacités d'approximation des métamodèles sur un exemple présentant une surface de réponse multimodale, on propose un nouveau cas-test dont la géométrie est proposée en figure 3.25. Ce cas-test illustre une opération d'assemblage par frettage d'un arbre dans son logement. La particularité de l'arbre considéré ici est qu'il présente un défaut de forme montré en figure 3.26.

	OK	OCK	OCK
n_s	9×9	3×3	5×5
Temps CPU moyen [s]	1.79s	2.17s	1.68s
Temps CPU mini. [s]	0.16s	0.16s	0.16s
Temps CPU maxi. [s]	8.01s	8.06s	8.08s
Nb. itérations total	6576	2711	5,666
Nb. itérations moyen	81.2	100	75.5
Nb. itérations mini.	5	5	5
Nb. itérations maxi.	381	381	381
Temps CPU total [s]	144s	58.5s	126s
Temps CPU sans MPS [s]	689s	225s	637s
Gain	4.78	3.84	5.06
R^2	**0.9986**	0.9756	**0.9955**
RAAE	$2.699 \cdot 10^{-2}$	0.1155	$4.468 \cdot 10^{-2}$
RMAE	0.1818	0.5352	0.3290
$\mathtt{Q}_1$	$2.068 \cdot 10^{-3}$	$1.793 \cdot 10^{-2}$	$6.774 \cdot 10^{-3}$
$\mathtt{Q}_2$	$3.631 \cdot 10^{-2}$	0.6087	0.1134
$\mathtt{Q}_3$	$\mathbf{9.078 \cdot 10^{-5}}$	$1.522 \cdot 10^{-3}$	$\mathbf{2.836 \cdot 10^{-4}}$

TABLE 3.13 – Krigeage et cokrigeage construits à partir de plans factoriels complets

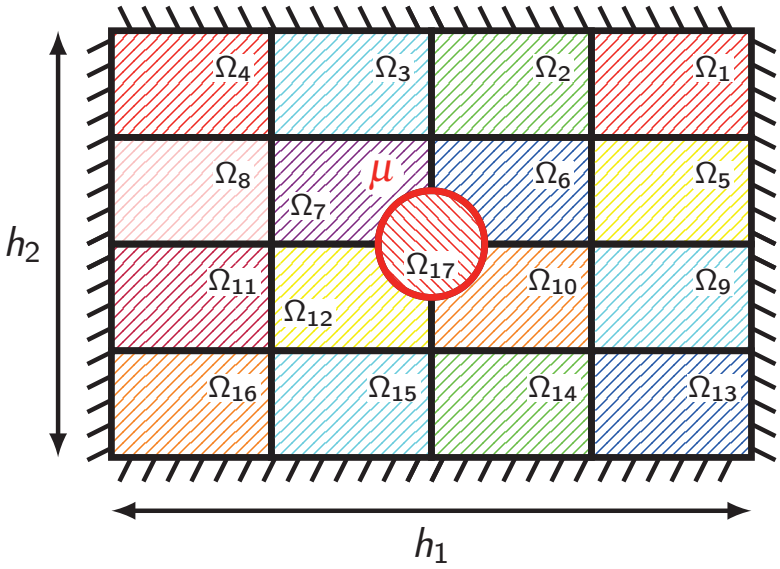


FIGURE 3.25 – Géométrie du problème de frettage

La résolution mécanique du cas-test consiste simplement à imposer de manière quasi-statique le défaut de l'arbre avant de lui appliquer un champ de pression vertical dont la valeur croît progressivement pour atteindre finalement $250\,\text{MPa}$. L'alésage ($h_1 = 0.3m$, $h_2 = 0.2m$, module de Young $E = 80GPa$, coefficient de Poisson $\nu = 0.3$) est composé de 16 sous-structures maillées en employant des triangles à 6 noeuds. L'arbre possède les mêmes caractéristiques que l'alésage et un diamètre nominal de $5\,\text{cm}$. Le défaut de forme présenté en figure 3.26 est imposé par l'intermédiaire d'un défaut d'excentricité $e = 0.5\,\mu\text{m}$ et d'une augmentation du diamètre de l'arbre de $1\,\mu\text{m}$.

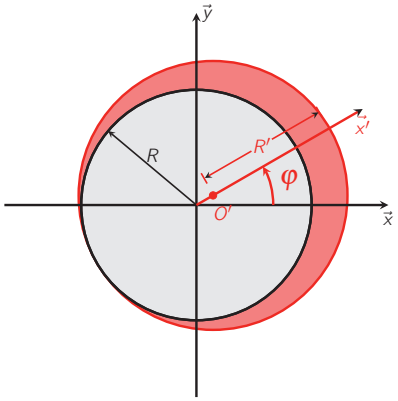


FIGURE 3.26 – Géométrie nominale et avec défaut de l'arbre

Pour l'étude envisagée ici, on considère deux paramètres que sont le coefficient de frottement μ entre l'arbre et l'alésage et l'angle d'orientation du défaut d'excentricité. On considère comme fonction objectif le maximum de la contrainte de Von Mises dans la structure.

La surface de réponse de référence obtenue à l'aide d'une grille régulière comportant 400 points (20×20) est présentée en figure 3.27. On peut remarquer qu'elle présente plusieurs minima locaux et un minimum global.

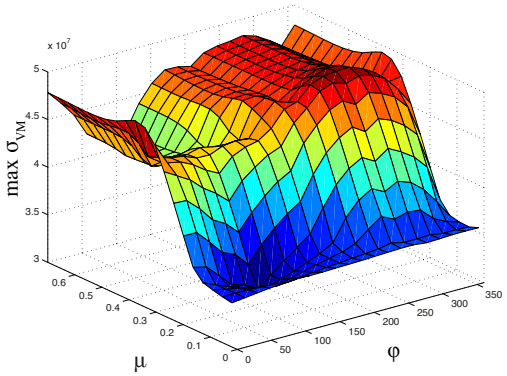
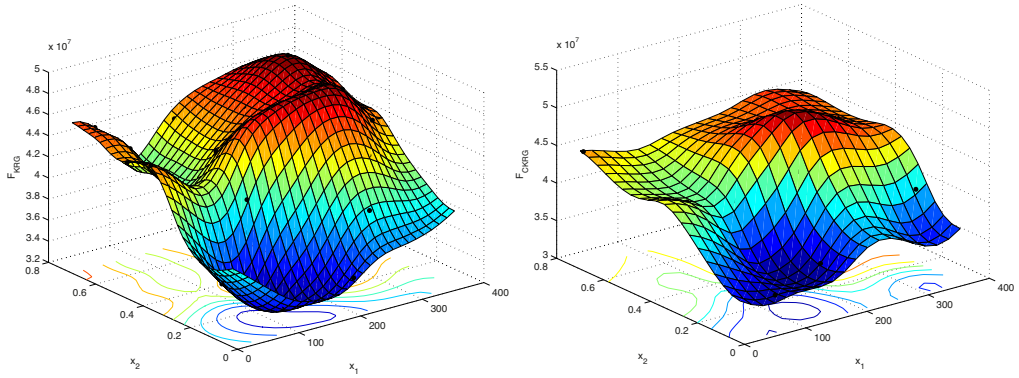


FIGURE 3.27 – Surface de réponse de référence du cas-test de frettage (contrainte de Von Mises maximale dans l'assemblage)

Les métamodèles de krigeage et cokrigeage ordinaire sont construits à partir de réponses et gradients obtenus en respectivement 30 et 10 points par l'intermédiaire une méthode LHS. La figure 3.28 et le tableau 3.14 montrent les approximations obtenues et les caractéristiques des calculs.



(a) Krigeage **(b)** Cokrigeage

FIGURE 3.28 – Approximations par krigeage et cokrigeage ordinaire obtenues à partir de 30 et 10 points (échantillonnage LHS)

Les métamodèles obtenus présentent des qualité similaires pour un gain quasiment équivalent. D'un point de vues résolution mécanique, cet exemple montre l'une des limites de la stratégie multiparamétrique. En effet, la rotation du défaut (obtenue par application d'un champ de jeu) engendre des changements dans la structure nécessitant un nombre important d'itérations à l'algorithme LATIN pour converger et par conséquent le temps de calcul augmente. Le gain n'est cependant pas négligeable puisque la stratégie multiparamétrique permet de réduire de moitié le temps de calcul nécessaire à la méthode LATIN pour obtenir les réponses sans la stratégie multiparamétrique.

5.2 Application à des exemples comportant plus de variables de conception

L'augmentation de la dimension du problème considéré engendrant une augmentation du nombre d'appels au solveur pour obtenir les gradients, le temps de calcul croît. Dans cette partie, on s'intéresse à la construction d'approximations par métamodèle avec et sans gradients sur un cas-test d'assemblage comportant 3 ou 4 variables. L'assemblage considéré est le cas-test CSMAb introduit dans la partie 4 page 130, dont la géométrie est rappelée dans la figure 3.29.

On considère les deux versions de ce cas-test dont les jeux de paramètres associés sont décrits dans la partie 4 page 130. L'étude proposée ici consiste à échantillonner l'espace de conception avec des tirages LHS comportant de 5 à 50 points, puis à construire, pour chaque jeu de paramètres, les approximations par krigeage ordinaire et RBF avec et sans

	KO	CKO
Number of evaluations	30	30
Number of responses	30	10 (10 eval.)
Number of gradients	0	10 (20 eval.)
Temps CPU moyen [s]	60.5s	45.6s
Temps CPU mini. [s]	33.1s	2.21s
Temps CPU maxi. [s]	123.6s	146.2s
Nb. itérations total	27137	19307
Nb. itérations moyen	905	644
Nb. itérations mini.	36	30
Nb. itérations maxi.	3605	2865
Temps CPU total [s]	1815s	1366s
Temps CPU sans MPS [s]	3641s	3441s
Gain	2.01	2.51
R^2	0.8841	0.8158
RAAE	0.2649	0.3367
RMAE	0.9956	1.183
$\mathtt{Q}_1$	$9.862 \cdot 10^{-3}$	$1.393 \cdot 10^{-2}$
$\mathtt{Q}_2$	0.4601	0.7312
$\mathtt{Q}_3$	$1.150 \cdot 10^{-3}$	$1.828 \cdot 10^{-3}$

TABLE 3.14 – Caractéristques des métamodèles et des calculs mécaniques du cas-test de frettage

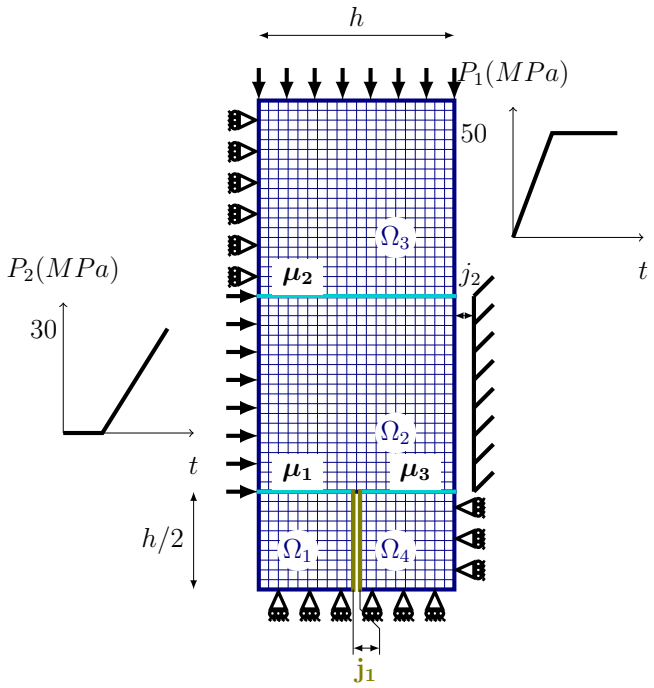
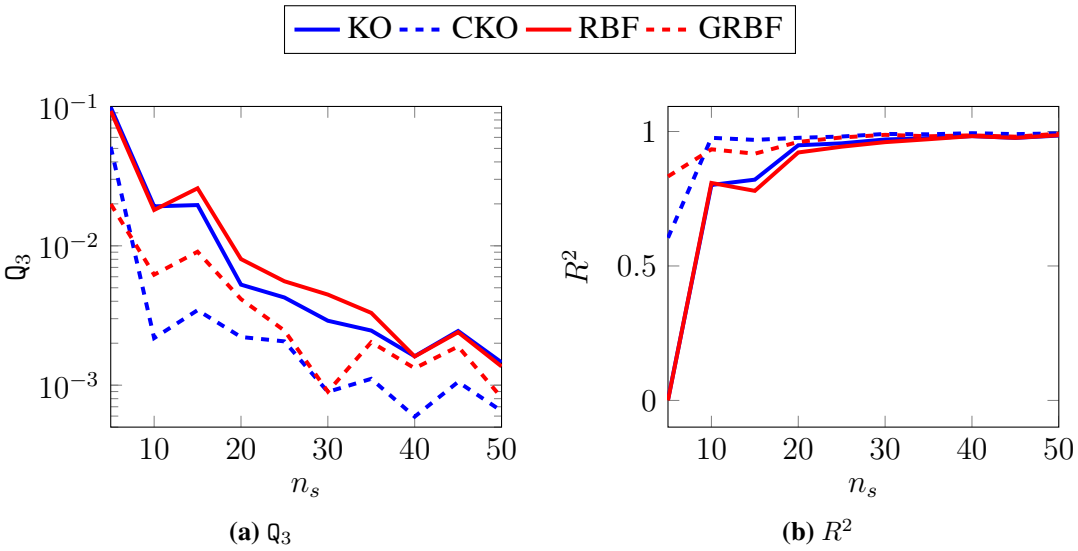


FIGURE 3.29 – Géométrie du cas-test CSMAb

gradients. Les valeurs des critères de qualité d'approximation des modèles de substitution sont regroupées dans les figures 3.30 et 3.31.

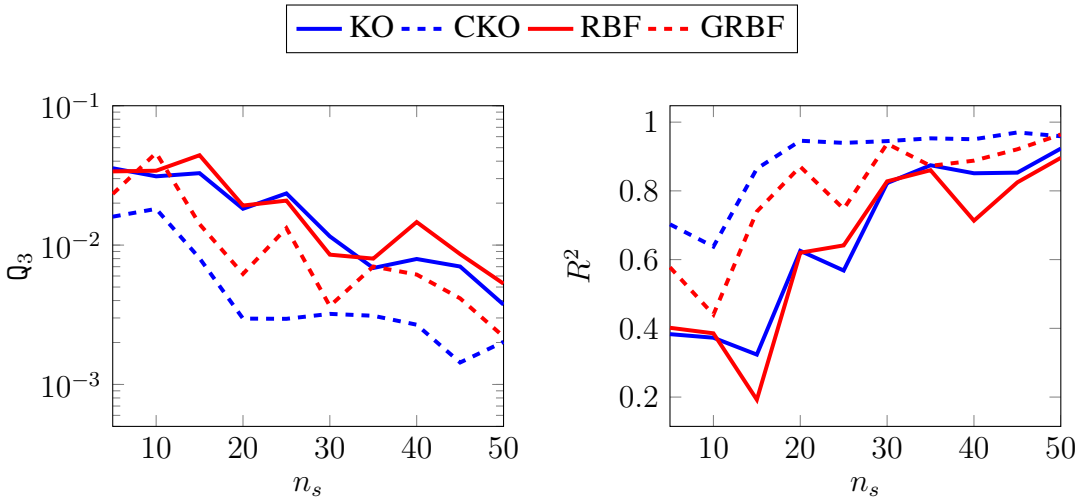


(a) Q_3 **(b)** R^2

FIGURE 3.30 – Approximations fournies pour le cas-test CSMAb à 3 paramètres

On observe sur ces résultats que :

► l'augmentation du nombre de points échantillonnés permet d'obtenir pour tous les



(a) Q_3 **(b)** R^2

FIGURE 3.31 – Approximations fournies pour le cas-test CSMAb à 4 paramètres

métamodèles une approximation de qualité ;

- lorsque le nombre d'échantillons est faible, les métamodèles à gradients fournissent une approximation de meilleure qualité.

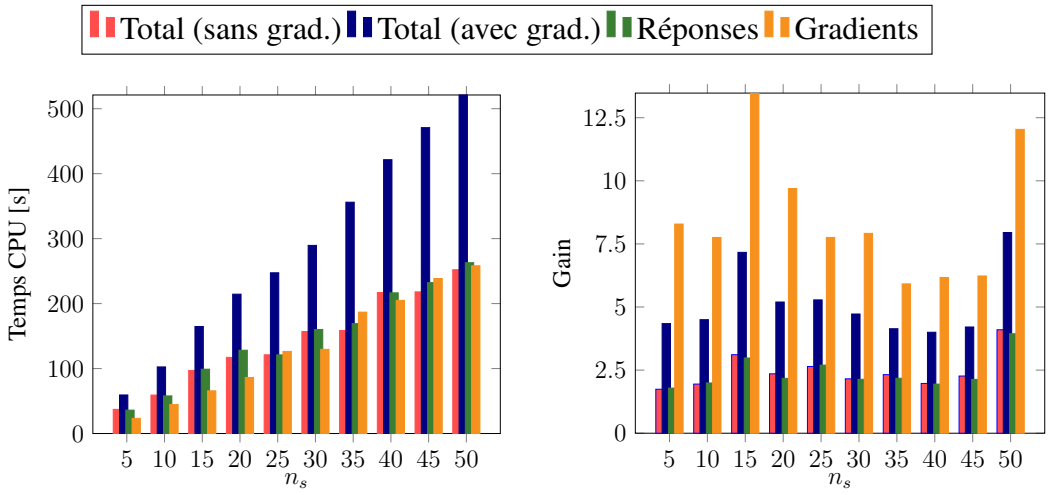
Ces deux constats confirment les éléments de résultats obtenus sur les fonctions analytiques. Par ailleurs, le modèle de substitution de type RBF à gradients (GRBF) fournit une approximation de moins bonne qualité à partir des mêmes échantillons. Ce phénomène est dû en grande partie au caractère multimodal de l'erreur de validation croisée utilisée pour déterminer les paramètres internes de ce métamodèle.

Dans un second temps, les temps de calcul associés à ces calculs sont présentés dans les figures 3.32 et 3.33.

L'emploi de la stratégie multiparamétrique permet une réelle diminution du coût de calcul :

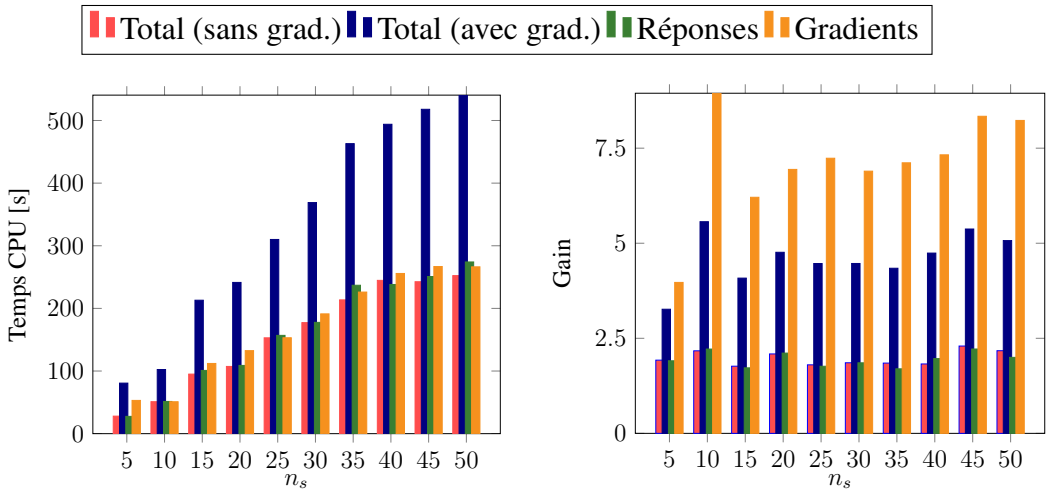
- dans les cas sans gradient, elle permet une diminution d'environ 80% du temps de calcul quel que soit le nombre d'échantillons ;
- le calcul des gradients est obtenu avec un gain particulièrement élevé (environ 7) ;
- sur les deux exemples l'accroissement du nombre de variables n'a pas d'influence sur le gain ;
- dans tous les cas, on constate que les gradients sont obtenus avec un temps de calcul équivalent à celui nécessaire à l'obtention des réponses.

Dans un contexte de bureau d'étude, où souvent le temps de calcul est limité, il s'avère important de pouvoir quantifier la qualité d'approximation atteinte en fonction



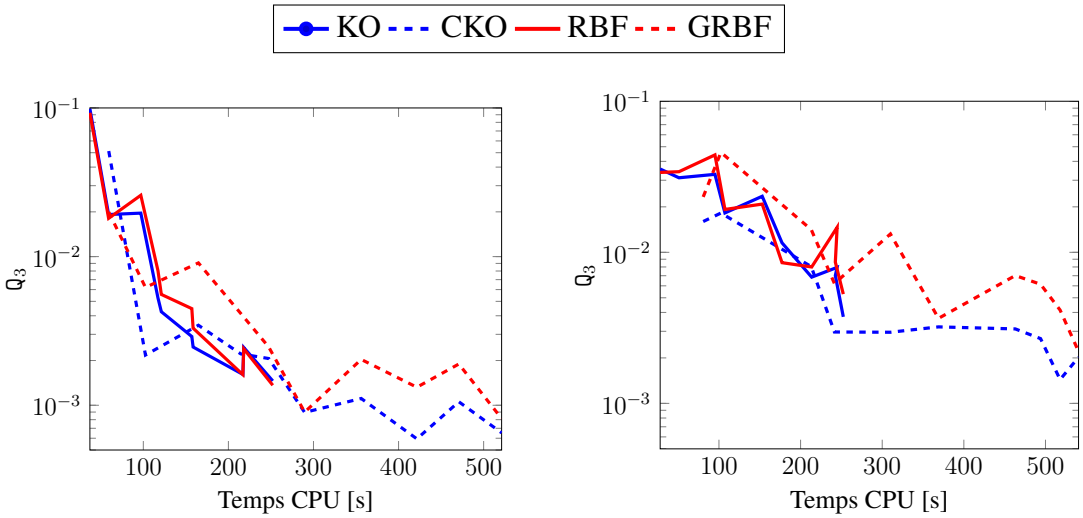
(a) Temps CPU [s] **(b)** Gains

FIGURE 3.32 – Temps de calcul et gains associés aux approximations dont les qualités sont données en figure 3.30 pour l'exemple CSMAb à 3 paramètres



(a) Temps CPU [s] **(b)** Gains

FIGURE 3.33 – Temps de calcul et gains associés aux approximations dont les qualités sont données en figure 3.31 pour l'exemple CSMAb à 4 paramètres



(a) CSMAb à 3 paramètres

(b) CSMAb à 4 paramètres

FIGURE 3.34 – Evolution de la qualité d'approximation en fonction du temps CPU pour les cas-tests CSMAb à 3 et 4 paramètres

du temps CPU. C'est pourquoi la figure 3.34 présente la qualité des approximations (par l'intermédiaire du critère Q_3) en fonction du temps CPU.

On observe alors que pour un temps CPU identique pour ces cas-tests, les métamodèles avec et sans gradients permettent d'obtenir des approximations de qualité similaire. Par ailleurs le modèle GRBF fournit, comme cela a été constaté précédemment, une approximation de qualité inférieure à celle des autres métamodèles.

Remarque 3.19 : Le caractère très variable de ces courbes est en grande partie dû au caractère aléatoire du tirage LHS utilisé. Une étude statistique de la qualité de ces approximations serait nécessaire pour confirmer précisément la tendance observée sur ces courbes.

6 Bilan de l'élaboration des modèles de substitution

Ce chapitre présente l'élaboration de métamodèles de type krigeage et RBF classiquement employés dans le cadre de l'optimisation. Après quelques rappels théoriques sur l'élaboration de ces deux types de métamodèles, leurs versions avec gradients ont été présentées dans le détail. Par ailleurs, des versions à gradients indirectes ont été proposées. Afin d'illustrer les qualités d'approximation de ces métamodèles, une étude complète sur des fonctions analytiques a été réalisée. Dans un dernier temps, la stratégie multiparamétrique a été employée dans le but de fournir réponses et gradients des fonctions objectifs considérées pour quelques cas-tests d'assemblages.

Au bilan, on constate que pour un même nombre de points échantillonnés, la prise en compte des gradients permet une réelle amélioration de la qualité d'approximation. L'étude

de qualité portant sur des fonctions analytiques a permis d'illustrer ces performances sur des fonctions plus ou moins complexes. Le cokrigeage s'est réellement détaché des autres méthodes par sa capacité à approcher des fonctions complexes que les autres approches ont du mal à approcher. Si l'on considère des exemples plus proches des fonctions objectifs issues de la mécanique, on s'aperçoit que krigeage et RBF permettent d'obtenir des approximations de qualité similaire. Cependant, la plus grande robustesse des méthodes basées sur le krigeage nous a amenés à ne retenir que ces approches. Dans un contexte mécanique, l'usage de la stratégie multiparamétrique permet de construire, sur les cas-tests d'assemblages proposés, des approximations de meilleure qualité pour un même nombre de points lorsque les gradients sont fournis (plus d'informations sont donc prises en compte lorsque l'on utilise les gradients). Finalement, l'observation de la qualité d'approximation sur deux exemples en fonction du temps CPU montre que l'usage de la stratégie multiparamétrique permet de construire des métamodèles de qualité similaire à temps CPU constant avec et sans gradients.

Les métamodèles de krigeage et cokrigeage apparaissent finalement comme des outils adaptés à l'optimisation d'assemblage. Ils vont alors pouvoir faire partie intégrante de la stratégie d'optimisation proposée dans le chapitre suivant.

Chapitre 4

Mise en œuvre d'une méthode d'optimisation à deux niveaux

Ce chapitre présente une méthode d'optimisation multiniveaux basée sur les éléments introduits dans les chapitres précédents. Le premier niveau, constitué d'un métamodèle, d'une procédure d'enrichissement et d'une optimisation globale, conduit à l'obtention d'un minimum approché. Le second niveau repose sur une optimisation locale et précise employant le modèle mécanique d'assemblage basé sur la stratégie multiparamétrique. La stratégie sera ensuite employée sur quelques cas-tests.

Sommaire

1 Description de l'approche considérée 202
2 Premier niveau 203
2.1 Construction du modèle de substitution 203
2.2 Optimisation globale sur le métamodèle 203
3 Second niveau 216
4 Application de la procédure d'optimisation à des assemblages 216
4.1 Etude de l'impact d'un rainurage sur une répartition de pression . . 216
4.2 Montage fretté 222
4.3 Joint d'accouplement 225
5 Bilan de la stratégie d'optimisation proposée 227

1 Description de l'approche considérée

La résolution d'un problème d'assemblage industriel qui comporte un nombre important de degrés de liberté, d'interfaces à comportement non-linéaire peut nécessiter plusieurs heures à plusieurs jours de calcul. Cette résolution est obtenue pour un jeu de paramètres choisi. Or, dans le cadre de l'optimisation, l'obtention du minimum global d'une fonction objectif associée à cette résolution peut nécessiter plusieurs centaines, voire plusieurs milliers de résolutions. C'est pour remédier à cette problématique de temps de calcul que cette approche bi-niveaux de modèles est proposée.

La démarche introduite ici et dont le principe est présenté en figure 4.1, s'insère dans le cadre de l'optimisation multiniveaux de modèles hiérarchique introduite dans la partie 2.4.3 du chapitre 1 page 37. Elle se structure autour de deux niveaux d'optimisation interagissant chacun avec un modèle donné. L'objectif de cette approche consiste à rendre abordable une optimisation globale en termes de temps de calcul. Pour cela, elle s'appuie sur deux éléments clés :

- l'emploi de la stratégie multiparamétrique qui permet, comme nous l'avons vu dans le chapitre 2, d'obtenir les réponses et gradients d'une fonction objectif issue de la résolution mécanique d'un problème d'assemblage pour un jeu de paramètres donné.
- l'usage d'un métamodèle à gradients capable, en utilisant les réponses et gradients calculés en un nombre limité de jeux de paramètres échantillonnés, de fournir une approximation de la fonction objectif de manière très peu coûteuse.

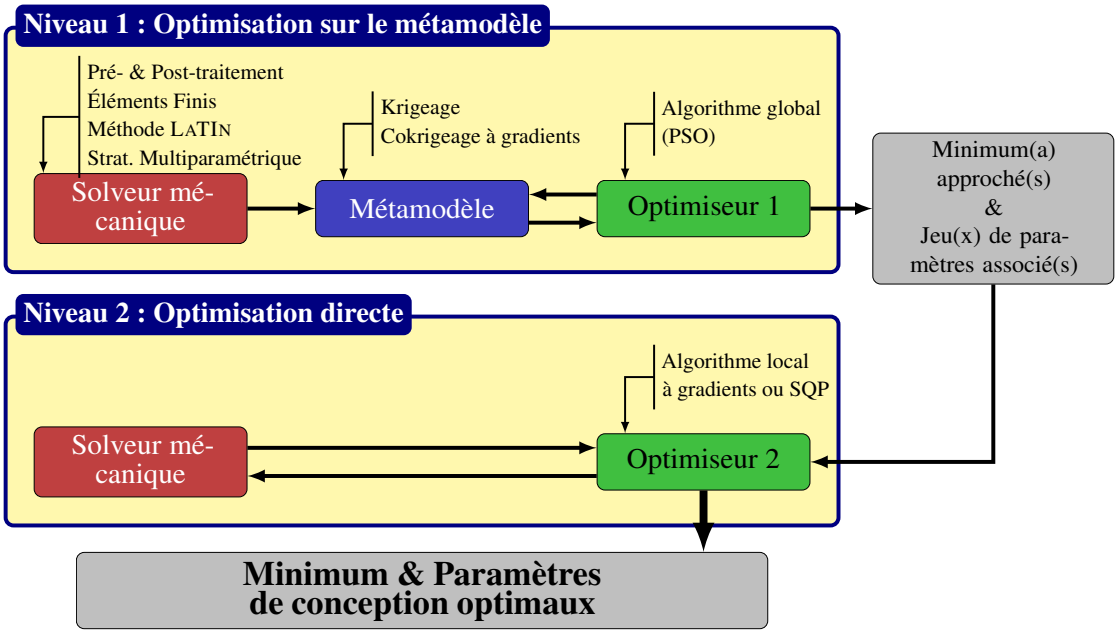


FIGURE 4.1 – Schéma synoptique de l'optimisation à deux niveaux de modèles

La démarche d'optimisation se déroule alors en deux phases :

- une optimisation globale sur le métamodèle permet de fournir une ou des zones potentielle(s) où se trouve le minimum ;
- une optimisation locale, qui emploie le solveur mécanique, initialisée avec les informations issues de l'optimisation globale.

On propose dans la suite de s'intéresser aux différents principes de la mise en œuvre des optimisations de chacun des niveaux. La démarche sera employée pour résoudre des problèmes d'optimisation portant sur différents types d'assemblages.

2 Premier niveau

Le premier niveau, constitué autour d'une optimisation globale, nécessite au préalable la construction du modèle de substitution. La problématique de cette partie est ici d'être capable de construire un métamodèle de qualité permettant de s'assurer de la localisation d'une ou de bonne(s) zone(s) potentielle(s) où se trouve le minimum global.

2.1 Construction du modèle de substitution

La première étape du processus d'optimisation globale consiste alors à construire le modèle de substitution. Suite aux chapitres précédents, on propose d'utiliser la stratégie multiparamétrique pour obtenir réponses et gradients de la fonction objectif. Ces évaluations seront obtenues en un ensemble de n_s jeux de paramètres échantillonnés obtenus par une méthode de tirages de type LHS ou IHS (cf. partie 4.4.2 page 70).

2.2 Optimisation globale sur le métamodèle

Une fois le métamodèle construit, l'idée de base consiste à réaliser l'optimisation globale directement sur le métamodèle. Cette phase, bien que simple à mettre en œuvre, est réalisée avec un coût très faible. Elle peut conduire à une convergence vers un minimum local existant sur le métamodèle mais inexistant pour la fonction réelle issue du solveur mécanique. Ce phénomène est dû à plusieurs éléments :

- l'échantillonnage, choisi initialement pour construire le métamodèle, peut ne pas s'avérer pertinent pour obtenir une approximation correcte dans la zone où se trouve le minimum ;
- la présence sur la fonction objectif réelle d'une « vallée », c'est-à-dire une zone où la fonction prend une valeur très proche de sa valeur minimale ;
- la présence de plusieurs minima globaux de même valeur.

La solution pour remédier à ces 3 problèmes consiste à employer un échantillonnage très dense dans la zone où se trouve le minimum. Or, dans la pratique, cette zone n'est pas connue. Il serait donc envisageable de densifier l'échantillonnage sur l'intégralité

de l'espace de conception. Cette méthode ne s'avère bien entendu pas viable car elle engendrerait un nombre d'appels au solveur mécanique bien trop important. La seule solution possible est alors d'utiliser le métamodèle pour obtenir la ou les zones nécessitant un ajout d'information. On parlera alors d'**enrichissement**. Ce dispositif fera l'objet de la partie suivante.

Par ailleurs, la méthode d'enrichissement pourra fournir une réponse au deuxième point introduit précédemment. La problématique de la présence de plusieurs minima ne sera pas abordée dans ce manuscrit. Elle nécessiterait une stratégie de traitement de chacun des minima pour s'assurer de leur pertinence dans la recherche du minimum global. Pour quelques éléments de réponse quant au traitement de ces candidats potentiels, on pourra se référer aux travaux suivants [VILLANUEVA et al. 2013].

2.2.1 Méthodes d'enrichissement

La nécessité de l'ajout d'une phase d'enrichissement de l'approximation peut parfaitement se justifier dans le cadre de la construction d'une approximation par modèle de substitution. En effet, dans ce cas, la pertinence et la qualité du modèle de substitution ne peuvent être connues qu'une fois ce dernier construit.

L'idée de base consiste à ajouter un nouveau jeu de paramètres parmi les échantillons préalablement tirés. Elle ne sera pas étudiée dans le cadre de l'enrichissement que l'on considérera ici. En effet, il peut s'avérer particulièrement difficile d'ajouter un nouveau jeu de paramètres sans altérer les propriétés de l'échantillonnage initial. De plus, cet ajout ne pourrait *a priori* se faire que dans une zone ayant un déficit de réponses. La pertinence du point obtenu vis-à-vis de l'amélioration de l'approximation fournie par le modèle de substitution serait alors discutable.

C'est dans le but de parvenir à ajouter des informations significatives au modèle de substitution qu'un certain nombre de travaux ont cherché à tirer profit d'informations fournies par le modèle de substitution pour l'enrichir de manière pertinente. Jones propose dans ces travaux l'emploi et la comparaison de plusieurs critères d'enrichissement classés en deux catégories [D. R. JONES 2001] :

- les approches à deux étapes (*Two-Stage*), pour lesquelles le modèle de substitution est tout d'abord construit à partir de l'échantillonnage initial, puis le critère d'enrichissement est élaboré à partir du métamodèle et est basé sur la réponse approchée ou la variance de l'estimateur de krigeage (par exemple). Le point obtenu par optimisation de ce critère vient s'ajouter à l'échantillonnage. Le processus peut alors être répété (construction du métamodèle et optimisation du critère).

- les approches à une étape (*One-Stage*), où le modèle de substitution n'est pas intégralement construit (l'approximation n'est pas utilisée) mais où les grandeurs mathématiques ou les principes de sa formulation sont utilisés.

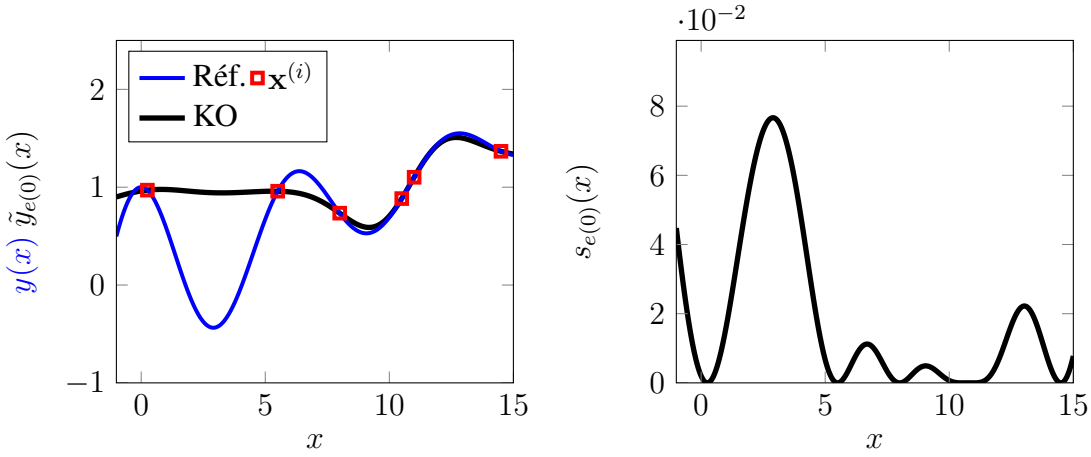
Dans le cadre de ces travaux, dans lesquels le coût de résolution du problème d'assemblage reste très important face au temps de construction du métamodèle, les approches

One-Stage ne seront pas étudiées. Nous nous concentrerons alors sur les méthodes *Two-Stage* dont les différents critères employables feront l'objet des parties suivantes.

Les notations utilisées seront du type $\bullet_{e(k)}$ ($\forall k \in \mathbb{N}$) pour désigner la grandeur $\bullet$ du métamodèle construite suite à l'ajout du k-ème point obtenu par enrichissement. Par exemple, l'approximation $\tilde{y}$ fournie par le métamodèle sera notée $\tilde{y}_{e(k)}$ telle que

- $\tilde{y}_{e(0)}$ soit l'approximation fournie par le métamodèle construit à partir des n_s réponses obtenues grâce à l'échantillonnage initial ;
- $\tilde{y}_{e(k)}$ soit l'approximation fournie par le métamodèle construit à partir des $n_s + k$ réponses aux n_s points $\mathbf{x}^{(i)}$ obtenus par l'échantillonnage initial et k nouveaux points $\mathbf{x}^{(n_s+k)}$.

Afin d'illustrer les différents critères proposés dans ce qui suit, on considère la fonction $y(x) = \exp(-x/10)\cos(x) + x/10$ introduite auparavant et un métamodèle de type krigeage ordinaire construit avec 6 réponses (cf. figure 4.2).



(a) Référence et krigeage ordinaire

(b) Variance du krigeage ordinaire

FIGURE 4.2 – Fonction d'exemple d'une variable et approximation par krigeage ordinaire

■ **Enrichissement par l'optimum :** Cette approche est une méthode pouvant s'appliquer à n'importe quel type de métamodèles. Elle se base sur une idée extrêmement intuitive consistant à choisir comme nouveau point celui correspondant au minimum du métamodèle. Ainsi, le nouveau point est défini tel que :

$$\forall k \in \mathbb{N}^*,\ \mathbf{x}^{(n_s+k)} = \arg\min_{\mathbf{x}\in\mathcal{D}} \tilde{y}_{e(k-1)}(\mathbf{x}) \tag{4.1}$$

On pourra trouver des exemples d'applications de cette méthode dans [LEARY, BHASKAR et KEANE 2004b ; PETER et al. 2007]. Bien qu'extrêmement simple à mettre en œuvre, elle n'est pas du tout adaptée aux fonctions complexes où l'ajout d'un nouveau point par cette méthode peut se traduire par une convergence vers un minimum local. La figure 4.3 illustre ce phénomène où la méthode converge vers un minimum local.

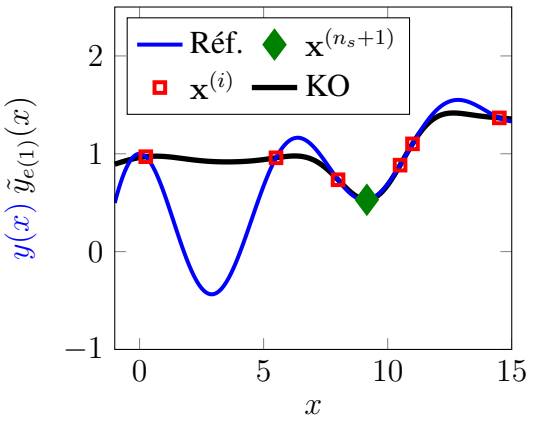


FIGURE 4.3 – Enrichissement par l'optimum et convergence locale (ajout d'un seul nouveau point)

■ **Enrichissement par la variance :** Lors de l'emploi d'un métamodèle de type krigeage, cokrigeage à gradients ou RBF sans gradient, ce dernier peut fournir une information d'ordre statistique quantifiant la variabilité de l'estimateur. Celui-ci, connu sous le nom de variance de l'estimateur ou erreur des moindres carrés, est défini en tout point de l'espace de conception et traduit directement le manque d'information dans certaines zones de l'espace des paramètres. Dans le cadre de l'usage de réponses déterministes, cette grandeur est nulle aux points échantillonnés et augmente avec l'éloignement de ceux-ci (voir par exemple la figure 3.3b page 148).

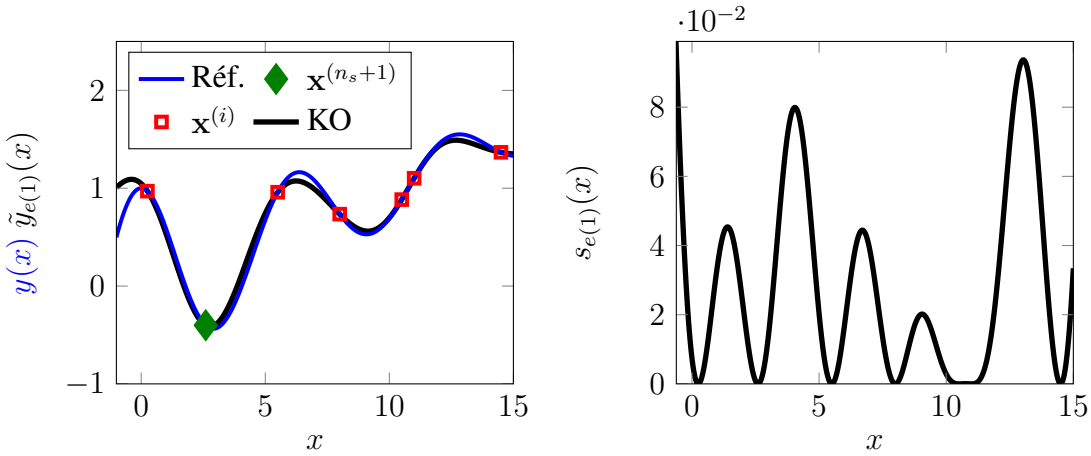
Le principe de l'enrichissement par la variance consiste à chercher le point de l'espace des paramètres associé à la variance maximale. Le point est alors défini de la manière suivante :

$$\forall k \in \mathbb{N}^*,\ \mathbf{x}^{(n_s+k)} = \arg\max_{\mathbf{x}\in\mathcal{D}} s_{e(k-1)}(\mathbf{x}) \tag{4.2}$$

Cette approche illustrée par la figure 4.4 permet d'obtenir à terme le minimum global. En effet, elle conduit à une densification de l'échantillonnage si l'on ajoute des points uniquement dans les zones qui en manquent le plus, ce qui permet alors d'obtenir d'un modèle de substitution précis sur l'ensemble de l'espace des paramètres.

Comme on l'observe sur la figure, l'ajout d'un nouveau site par variance maximale peut permettre d'obtenir une meilleure localisation du minimum. Sur le cas présenté, il aura suffi d'un seul point supplémentaire, mais sur une fonction plus complexe ou lors de l'optimisation d'une fonction comportant plus de variables, on ne peut être assuré d'une convergence aussi rapide.

■ **Enrichissement par compromis optimum/variance :** Les deux approches précédentes présentent toutes deux des inconvénients majeurs qui les rendent inexploitables en pratique. Cependant, il est envisageable de trouver un compromis entre ces deux approches en cherchant à exploiter au maximum les informations issues du métamodèle (c'est clairement ce que propose l'enrichissement par l'optimum de la fonction) et à explorer au maximum les zones où l'incertitude sur la fonction est la plus grande (c'est ce que



(a) **(b)**

FIGURE 4.4 – Enrichissement par la variance maximale : approximation et variance du krigeage ordinaire (ajout d'un seul nouveau point) à comparer avec la figure 4.2

propose le critère d'enrichissement par le maximum de la variance). On a alors affaire à un compromis **exploration/exploitation**.

Une approche envisageable consisterait à chercher un nouveau point solution du problème à deux critères :

$$\min_{\mathbf{x}\in\mathcal{D}} \tilde{y}_{e(k-1)}(\mathbf{x}) \text{ et } \max_{\mathbf{x}\in\mathcal{D}} s_{e(k-1)}(\mathbf{x}) \tag{4.3}$$

La résolution de ce problème pourrait employer le principe du front de Pareto. Il demeure qu'un nombre extrêmement important de solutions à ce problème existe néanmoins. En trouver et en choisir une reste un problème complexe. Par ailleurs, le coût nécessaire à cette résolution devient important avec la taille du problème. On pourra se référer à [D. R. JONES, PERTTUNEN et STUCKMAN 1993] pour un exemple d'application simple de cette approche.

■ **Enrichissement par intervalle de confiance :** Connue sous le nom *Lower Confidence Bound* (LCB) et introduite par [COX et JOHN 1997], cette approche consiste à chercher le point d'enrichissement tel que

$$\forall k \in \mathbb{N}^*,\ \mathbf{x}^{(n_s+k)} = \arg\min_{\mathbf{x}\in\mathcal{D}} \text{LCB}(\mathbf{x}) \text{ où } \text{LCB}(\mathbf{x}) = \tilde{y}_{e(k-1)}(\mathbf{x}) - b s_{e(k-1)}(\mathbf{x}) \tag{4.4}$$

La valeur de b permet d'ajuster l'intervalle de confiance considéré et par ce biais de privilégier exploration ou exploitation. On choisit des valeurs de b faibles pour privilégier la fonction au détriment de la variance (exploitation) et, à l'inverse, choisir une valeur de b élevée permet de privilégier les zones peu explorées. Contrairement à l'approche précédente, ce critère est beaucoup plus simple à mettre en œuvre en raison de la minimisation d'une seule fonction.

■ **Enrichissement basé sur la probabilité d'amélioration :** La méthode proposée ici consiste à utiliser le caractère aléatoire du krigeage. On cherchera à maximiser la probabilité d'amélioration (*Probability of Improvement* PI) de la fonction [KUSHNER 1964], c'est-à-dire la probabilité qu'elle soit inférieure à une valeur donnée T. Pour cela, on établit la probabilité d'amélioration en considérant que la valeur de l'estimation est une réalisation d'une variable aléatoire de moyenne $\tilde{y}_{e(k)}(\mathbf{x})$ et d'écart-type $s_{e(k)}(\mathbf{x})$ de la manière suivante :

$$\mathrm{PI}(\mathbf{x}) = P(\hat{Y} < T) = \Phi\left(\frac{T - \tilde{y}_{e(k)}(\mathbf{x})}{s_{e(k)}(\mathbf{x})}\right) \tag{4.5}$$

où Φ est la fonction de répartition de la loi normale centrée réduite ($\Phi(u) = \frac{1}{2}(1 + \mathrm{err}(u))$). On pourra par exemple choisir de maximiser la probabilité d'obtenir un meilleur minimum que ceux obtenus aux points échantillonnés en fixant $T = y_{\min} = \min(y_i, i \in [\![1, n_s]\!])$ ou chercher une probabilité d'amélioration de 25% en choisissant $T = y_{\min} - 0.25|y_{\min}|$.

Le choix d'un valeur pertinente de T est extrêmement complexe. En effet, si l'amélioration souhaitée est trop faible, alors la recherche du point d'enrichissement sera extrêmement locale et ne deviendra exploratoire qu'à partir du moment où le voisinage du point $y_{\min}$ aura été exploré en détail. A l'inverse, fixer une amélioration souhaitée trop importante engendrera une recherche très globale et la convergence de l'optimisation globale sera alors très lente. Pour remédier à ce problème, la solution pourrait être de choisir plusieurs valeurs de T pour en déduire plusieurs points d'enrichissement [D. R. JONES 2001] (cet article contient également des compléments en termes d'études et de références sur ce critère).

■ **Enrichissement basé sur l'amélioration espérée :** Le critère précédent permet d'évaluer la probabilité d'amélioration sans la quantifier. C'est dans ce but que le critère d'amélioration espérée (*Expected Improvement* EI) a été envisagé [SCHONLAU 1997]. Il se base sur le même contexte statistique que le critère précédent où la valeur de l'estimation est la réalisation d'une variable aléatoire $Y(\mathbf{x})$ normalement distribuée, de moyenne $\tilde{y}_{e(k)}(\mathbf{x})$ et d'écart-type $s_{e(k)}(\mathbf{x})$.

L'amélioration au point $\mathbf{x}$ peut être formulée telle que $\mathrm{I}(\mathbf{x}) = \max(y_{\min} - Y(\mathbf{x}))$. De cette grandeur aléatoire, on détermine l'amélioration espérée définie telle que

$$\mathrm{EI}(x) = \mathbb{E}[\mathrm{I}(\mathbf{x})] = \mathbb{E}[\max(y_{\min} - Y(\mathbf{x}))] \tag{4.6}$$

Le développement de cette expression conduit à (cf. [GINSBOURGER 2009] par exemple) :

$$\mathrm{EI}(x) = \left(y_{\min} - \tilde{y}_{e(k)}(\mathbf{x})\right)\Phi(z(\mathbf{x})) + s_{e(k)}(\mathbf{x})\phi(z(\mathbf{x})) \text{ avec } z(\mathbf{x}) = \frac{y_{\min} - \tilde{y}_{e(k)}(\mathbf{x})}{s_{e(k)}(\mathbf{x})} \tag{4.7}$$

où $\phi(u) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}u^2)$ et $\Phi(u) = \frac{1}{2}(1 + \mathrm{err}(u))$ sont respectivement la densité de probabilité et la fonction de répartition de la loi normale centrée réduite. Ce critère propose un compromis entre exploration et exploitation par le biais des deux termes qui le composent. Par ailleurs, il présente des propriétés qui rendent son usage extrêmement

intéressant dans le cadre de l'optimisation : le critère est nul aux points échantillonnés (ce qui permet d'éviter l'ajout de points existants) et est toujours positif.

Finalement, le nouveau point d'enrichissement est obtenu tel que

$$\forall k \in \mathbb{N}^*,\ \mathbf{x}^{(n_s+k)} = \arg\max_{\mathbf{x}\in\mathcal{D}} \mathrm{EI}(\mathbf{x}) \tag{4.8}$$

Cette méthode d'enrichissement a été intégrée dans l'approche EGO (*Efficient Global Optimization*) proposée par [D. R. JONES, SCHONLAU et WELCH 1998] et basée sur le krigeage.

La figure 4.5 présente l'enrichissement successif d'un métamodèle par l'amélioration espérée. Chaque graphique présente l'approximation et la valeur de l'amélioration espérée à l'itération courante. La recherche du maximum de l'amélioration espérée conduit à l'obtention d'un point qui sera utilisé pour construire le métamodèle de l'itération suivante. On observe que l'usage de l'amélioration espérée permet dans ce cas une convergence rapide vers le minimum global. Dès l'ajout du troisième point (figure 4.5d), la valeur de l'amélioration espérée chute très fortement et la procédure peut s'arrêter.

Par ailleurs, la très forte multimodalité de l'amélioration espérée nécessite l'usage d'un algorithme d'optimisation globale qui peut nécessiter un nombre important d'itérations pour converger. On rappelle que le calcul de EI ne dépend que des grandeurs fournies à coût très faible par le métamodèle (approximation et variance).

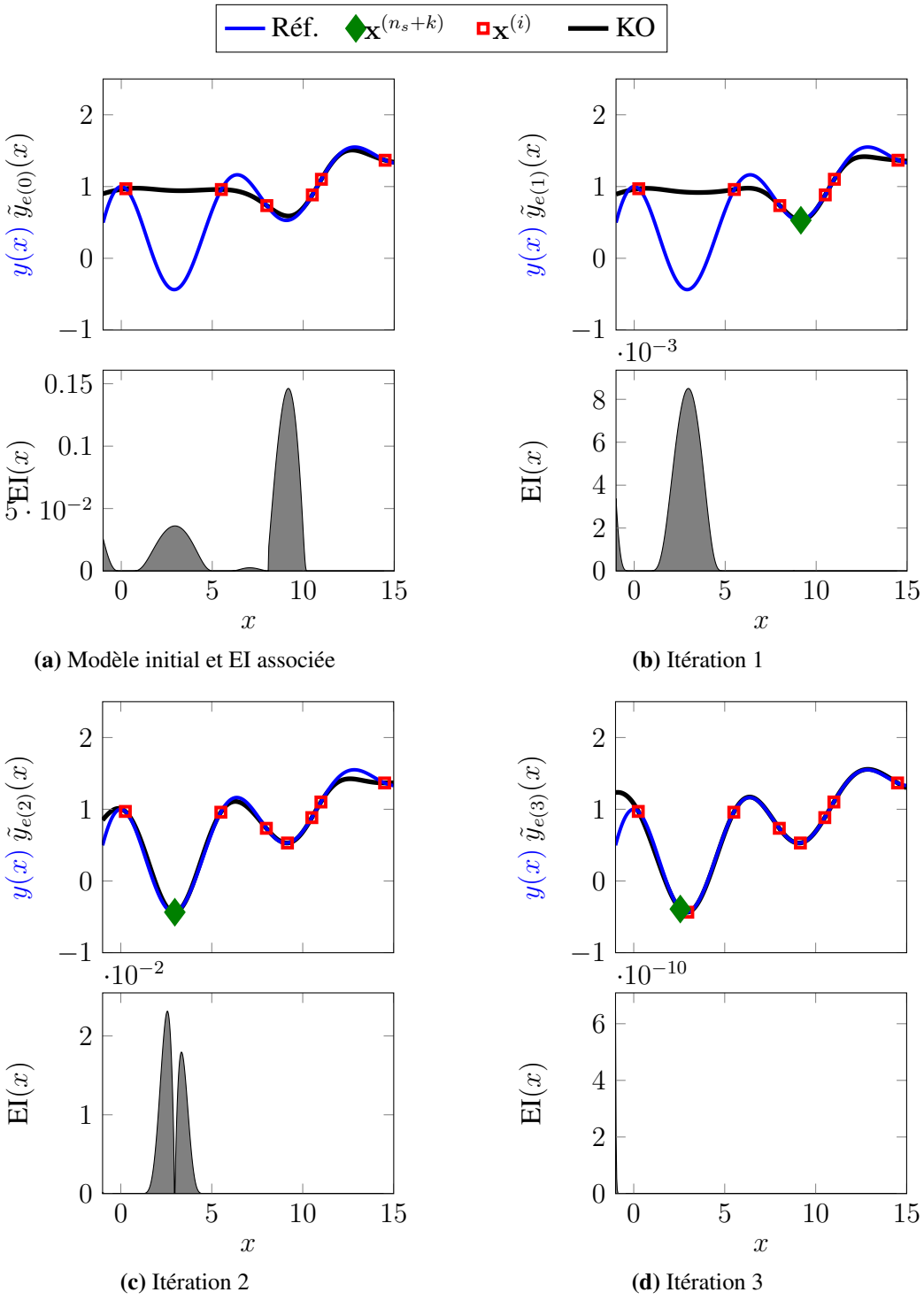
■ **Extensions de l'enrichissement basé sur l'amélioration espérée :** Le critère EI a connu au fur et à mesure des années de nombreux développements et extensions. Parmi ceux-ci, on pourra citer :

- l'amélioration espérée pondérée [SÓBESTER, LEARY et KEANE 2005] permettant de palier aux problèmes de convergence lente lorsque le tirage initial conduit à l'obtention d'un écart-type faible. Le principe de cette approche consiste à ajouter un coefficient permettant de privilégier l'exploration ou l'exploitation. Le critère s'exprime alors :

$$\forall w \in [0,1],\ \mathrm{WEI}(x) = w\left(y_{\min} - \tilde{y}_{e(k)}(\mathbf{x})\right)\Phi(z(\mathbf{x})) + (1-w)s_{e(k)}(\mathbf{x})\phi(z(\mathbf{x}))$$
$$\text{avec } z(\mathbf{x}) = \frac{y_{\min} - \tilde{y}_{e(k)}(\mathbf{x})}{s_{e(k)}(\mathbf{x})} \tag{4.9}$$

  On constate que $w = 0.5$ conduit à l'emploi du critère EI ($\mathrm{WEI}(x) = 0.5\mathrm{EI}(x)$) et que $w = 0$ et $w = 1$ conduisent respectivement à des critères d'exploration pure et d'exploitation pure. Suite à leurs travaux, SÓBESTER, LEARY et KEANE recommandent une approche cyclique consistant à changer la valeur de w au fur et à mesure des itérations en les choisissant successivement parmi $[0;1;0;3;0;5;0;7;0;9]$. Ce choix assure une exploration continue de l'espace permettant d'éviter une convergence trop lente.

- l'amélioration espérée généralisée [SCHONLAU 1997] qui permet de palier aux situations où la fonction objectif ne se laisse pas bien modéliser comme réalisation



(a) Modèle initial et EI associée

(b) Itération 1

(c) Itération 2

(d) Itération 3

FIGURE 4.5 – Enrichissement successif d'un métamodèle de krigeage ordinaire

d'un processus Gaussien et où l'EI fournit une information trop locale. Le critère associé se formule de la manière suivante :

$$\forall g \in \mathbb{N},\ \mathrm{EI}^g(\mathbf{x}) = s_{e(k)}(\mathbf{x}) \sum_{k=0}^{g} (-1^k) \left(\frac{g!}{k!(g-k)!} z(\mathbf{x})^{g-k} T_k \right) \tag{4.10}$$

où $T_k = -\phi(z(\mathbf{x}))z(\mathbf{x})^{k-1} + (k-1)T_{k-2}$, $T_0 = \Phi(z(\mathbf{x}))$, $T_1 = -\phi(z(\mathbf{x}))$ et $z(\mathbf{x}) = \dfrac{y_{\min} - \tilde{y}_{e(k)}(\mathbf{x})}{s_{e(k)}(\mathbf{x})}$.

On constate que pour $g = 1$ on retrouve le critère EI.

On trouve également des versions parallèles de l'EGO [GINSBOURGER 2009] ou adaptées aux métamodèles non-déterministes [HUANG et al. 2006].

■ **Autres approches d'enrichissement :** D'autres méthodes d'enrichissement peuvent être trouvées dans [JIN, W. CHEN et SUDJIANTO 2002] (basées sur une distance maximin modifiée et sur la validation croisée), [SASENA 2002] (basées sur des modifications ou des combinaisons des critères précédents) et [VILLEMONTEIX, VAZQUEZ et WALTER 2009] (critère SUR utilisant une quantification de l'information basée sur l'entropie).

■ **Éléments de comparaison :** On pourra se référer aux travaux [D. R. JONES 2001 ; SASENA 2002 ; J. LIU, Z.-H. HAN et SONG 2012] pour quelques éléments de comparaison de ces approches.

2.2.2 Mise en œuvre de l'optimisation globale

Afin de mener à terme cette phase d'optimisation globale basée sur le modèle de substitution, il a été choisi d'employer la méthode d'enrichissement basée sur l'amélioration espérée (EI). Classiquement, lors de la mise en œuvre d'un processus d'optimisation, la question de la mise en place d'un critère d'arrêt se pose. En absence de valeur connue du vrai minimum, ce critère est généralement basé sur l'évolution de la valeur du minimum atteint ou de celle du gradient. Dans le cadre de l'EGO, D. R. JONES, SCHONLAU et WELCH proposent l'usage du critère [D. R. JONES, SCHONLAU et WELCH 1998] :

$$\max_{\mathbf{x} \in \mathcal{D}} \frac{\mathrm{EI}(\mathbf{x})}{|y_{\min}|} < \delta \tag{4.11}$$

C'est-à-dire que l'algorithme s'arrête dès que la valeur maximale de l'EI sur l'espace des paramètres devient inférieure à la valeur $\delta|y_{\min}|$. La valeur classiquement choisie pour δ est 1%.

HUANG et al. proposent d'employer le critère suivant permettant de s'affranchir des problèmes liés à l'amplitude des évaluations [HUANG et al. 2006] :

$$\max_{\mathbf{x} \in \mathcal{D}} \frac{\mathrm{EI}(\mathbf{x})}{y_{\max} - y_{\min}} < \delta_r \tag{4.12}$$

où $y_{\max} = \max_i\ y_i$ et $y_{\min} = \min_i\ y_i$. On choisit généralement $\delta_r = 10^{-3}$.

En complément de ces critères d'arrêt, il est également possible de stopper l'algorithme en considérant un temps de calcul maximum pour le problème mécanique ou un nombre maximum d'appels au solveur. L'algorithme peut également être arrêté lorsque le nombre d'itérations de ce dernier dépasse une limite fixée par l'utilisateur.

Dans le cadre de l'usage de l'EGO, la construction du krigeage faisant intervenir de nouvelles informations (réponses et gradients) au nouveau point obtenu par enrichissement, il est très fortement recommandé de recalculer les paramètres du krigeage à chaque itération. En effet, comme le montre GINSBOURGER, ces paramètres peuvent varier fortement lorsque l'échantillon pour construire le métamodèle comporte peu d'éléments (voir [GINSBOURGER 2009, tableau 4.1 p. 118]).

L'emploi d'une démarche d'enrichissement performante comme celle basée sur l'amélioration espérée permet de ne considérer initialement qu'un échantillonnage de l'espace des paramètres comportant peu de points. D. R. JONES, SCHONLAU et WELCH conseillent de considérer un tirage initial de $10n_p$ points [SCHONLAU 1997 ; D. R. JONES, SCHONLAU et WELCH 1998].

2.2.3 Exemples d'application sur une fonction analytique

Afin d'illustrer les performances et l'évolution de grandeurs du métamodèle lors du déroulement de l'optimisation globale, l'EGO est employé pour trouver le minimum global d'une fonction test analytique introduite dans la partie 4.2 page 178 du chapitre 3 : la fonction *Six-Hump Camel Back*. Le protocole consiste à considérer un échantillonnage initial de type plan factoriel complet à 3 niveaux (soit une grille régulière 3×3) puis à mettre en œuvre l'algorithme en utilisant des métamodèles de krigeage et cokrigeage à gradients. La détermination des paramètres du krigeage et du cokrigeage sera réalisée en employant un algorithme PSO. Il en sera de même pour rechercher le maximum de l'amélioration espérée.

Les tableaux 4.1 et 4.2 présentent l'évolution des grandeurs lors de l'optimisation.

Bien que cette fonction comporte 2 minima globaux, on constate que les deux approches convergent vers un des deux minima globaux. En effet, le minimum approché fourni par les métamodèles à chaque itération converge en position ($\mathbf{x}^*$) et en amplitude ($\tilde{y}(\mathbf{x}^*)$) vers les vrais minima globaux. L'usage d'un métamodèle à gradients permet une location plus rapide du minimum (moins d'itérations). Ce constat s'explique par la meilleure qualité d'approximation fournie par le cokrigeage à gradients pour un même nombre d'itérations (cf. figures 4.7 et 4.8 à comparer avec la référence donnée par la figure 4.6). Ce modèle de subsitution nécessite cependant l'apport à chaque itération des réponses et des gradients de la fonction objectif, ce qui augmente le coût de construction de l'approximation.

Par ailleurs, dans le cas du krigeage, le recalcul des paramètres du krigeage à chaque itération apparaît comme indispensable (du moins pour les premières itérations). De plus, sur cette fonction test, le critère d'arrêt proposé par [HUANG et al. 2006] permet de stopper judicieusement l'algorithme. Cette étude ne permet cependant pas de tirer une conclusion nette quant à sa capacité de stopper l'algorithme de façon pertinente dans tous les cas de

Ite.	l		$\mathbf{x}^*$		$\tilde{y}(\mathbf{x}^*)$	$\mathrm{EI}_{\max}$	$\mathrm{EI}^j_{\max}$ [a]	$\mathrm{EI}^h_{\max}$ [b]
1	19.949	0.5	0	0.694	$-4.8 \cdot 10^{-2}$	0.333	0.333	$5.8 \cdot 10^{-2}$
2	5.855	0.5	$-8.8 \cdot 10^{-2}$	-0.56	-0.975	0.363	0.424	$5.5 \cdot 10^{-2}$
3	1.631	0.515	$-2.7 \cdot 10^{-2}$	-0.55	-0.866	0.191	0.223	$2.9 \cdot 10^{-2}$
4	0.984	0.5	$4.5 \cdot 10^{-2}$	-0.56	-0.904	$8.2 \cdot 10^{-2}$	$9.6 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$
5	0.892	0.5	$4.5 \cdot 10^{-2}$	-0.558	-0.904	$7.9 \cdot 10^{-2}$	$9.2 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$
6	0.822	0.5	$5 \cdot 10^{-3}$	-0.585	-0.875	$3.4 \cdot 10^{-2}$	$4 \cdot 10^{-2}$	$5 \cdot 10^{-3}$
7	0.671	0.5	$6.6 \cdot 10^{-2}$	-0.656	-0.991	$2.1 \cdot 10^{-2}$	$2.1 \cdot 10^{-2}$	$3 \cdot 10^{-3}$
8	0.625	0.5	0.11	-0.682	-1.022	$8 \cdot 10^{-3}$	$8 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
9	0.573	0.5	0.11	-0.682	-1.022	$1.5 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$2 \cdot 10^{-3}$
10	0.537	0.5	0.107	-0.683	-1.022	$1 \cdot 10^{-2}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$

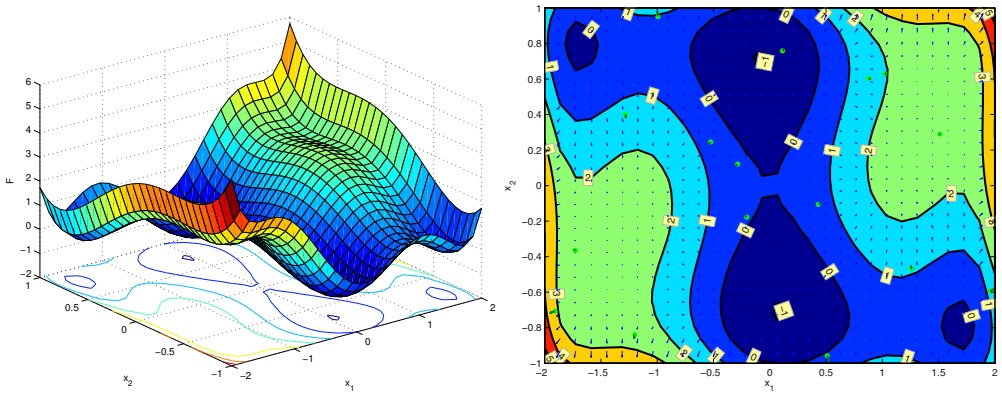
[a] Critère proposé dans [D. R. JONES, SCHONLAU et WELCH 1998]
[b] Critère proposé dans [HUANG et al. 2006]

TABLE 4.1 – Evolution des grandeurs lors de l'optimisation par EGO basée sur le krigeage ordinaire

Ite.	l		$\mathbf{x}^*$		$\tilde{y}(\mathbf{x}^*)$	$\mathrm{EI}_{\max}$	$\mathrm{EI}^j_{\max}$ [a]	$\mathrm{EI}^h_{\max}$ [b]
1	0.88	0.5	1.192	-0.741	-2.755	2.811	2.811	0.49
2	0.954	0.5	-1.209	0.743	-2.583	2.671	2.671	0.466
3	1.001	0.5	$-5 \cdot 10^{-2}$	0.695	-1.005	1.012	1.012	0.176
4	0.974	0.5	$-8 \cdot 10^{-2}$	0.71	-1.03	0.272	0.267	$4 \cdot 10^{-2}$
5	0.937	0.5	$-7.8 \cdot 10^{-2}$	0.709	-1.029	0.225	0.22	$3.3 \cdot 10^{-2}$
6	0.903	0.5	$-7.8 \cdot 10^{-2}$	0.706	-1.028	0.178	0.174	$2.6 \cdot 10^{-2}$
7	0.867	0.5	$-7.8 \cdot 10^{-2}$	0.706	-1.028	$8.8 \cdot 10^{-2}$	$8.6 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$
8	0.82	0.5	$-8.9 \cdot 10^{-2}$	0.707	-1.03	$5.9 \cdot 10^{-2}$	$5.8 \cdot 10^{-2}$	$9 \cdot 10^{-3}$
9	0.787	0.5	$-8.9 \cdot 10^{-2}$	0.707	-1.03	$3.3 \cdot 10^{-2}$	$3.3 \cdot 10^{-2}$	$5 \cdot 10^{-3}$
10	0.788	0.5	$-9.1 \cdot 10^{-2}$	0.708	-1.031	$3.2 \cdot 10^{-2}$	$3.1 \cdot 10^{-2}$	$5 \cdot 10^{-3}$

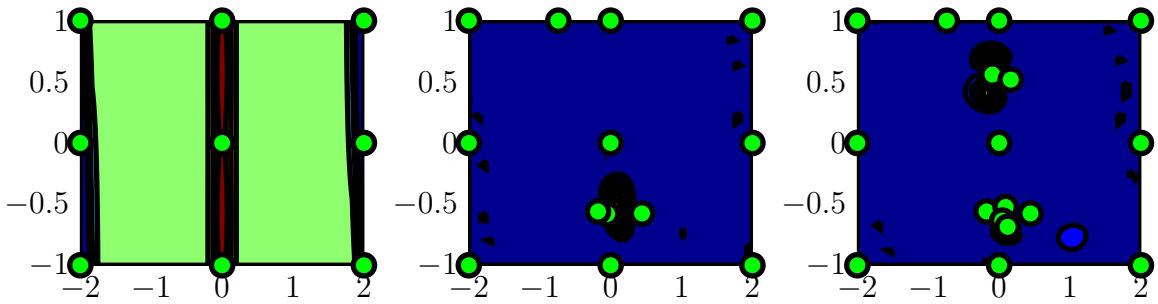
[a] Critère proposé dans [D. R. JONES, SCHONLAU et WELCH 1998]
[b] Critère proposé dans [HUANG et al. 2006]

TABLE 4.2 – Evolution des grandeurs lors de l'optimisation par EGO basée sur le cokrigeage ordinaire à gradients

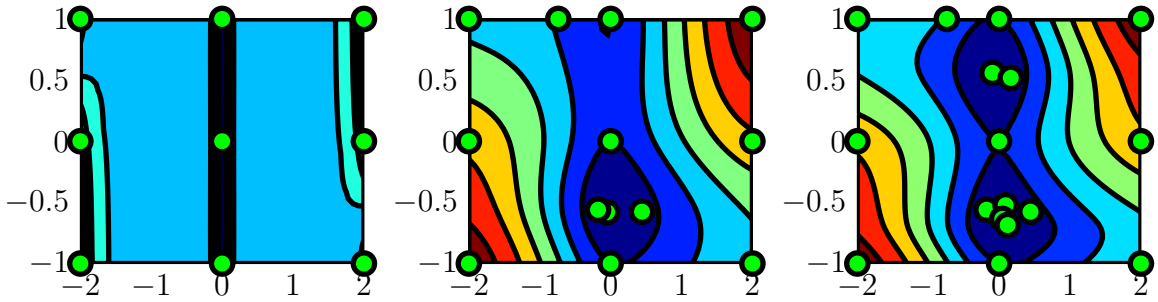


(a) Surface de réponse **(b)** Courbes de niveaux

FIGURE 4.6 – Fonction *Six-Hump Camel Back*

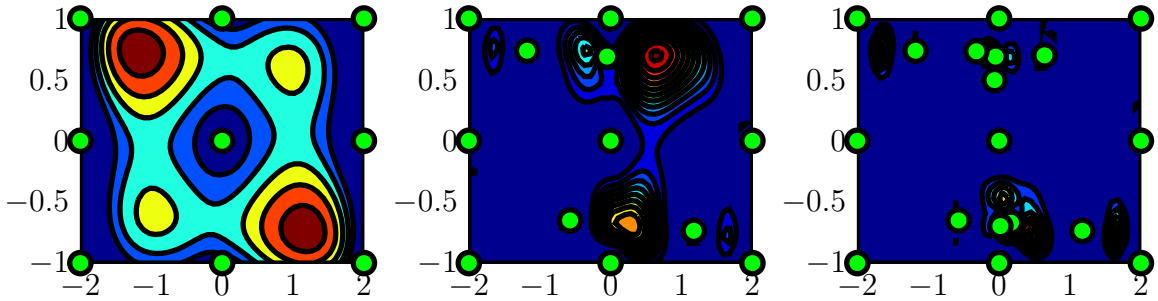


(a) Amélioration espérée

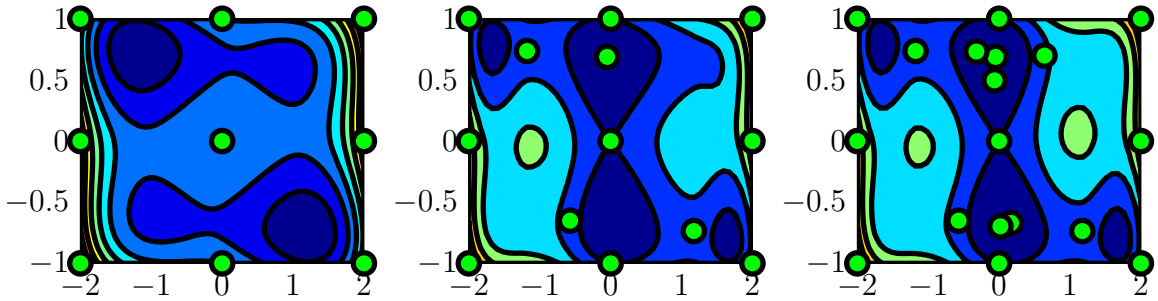


(b) Approximation par krigeage ordinaire

FIGURE 4.7 – Allure des surfaces de réponses de l'amélioration espérée et de l'approximation fournie par krigeage ordinaire aux itérations 1, 5 et 10 de l'algorithme EGO



(a) Amélioration espérée



(b) Approximation par cokrigeage ordinaire à gradients

FIGURE 4.8 – Allure des surfaces de réponses de l'amélioration espérée et de l'approximation fournie par cokrigeage ordinaire à gradients aux itérations 1, 5 et 10 de l'algorithme EGO

figure. En outre, les minima atteints sont proches des véritables minima mais l'algorithme nécessiterait encore quelques itérations pour mieux converger. C'est pourquoi, par la suite, il semble préférable d'ajouter une optimisation plus fine pour localiser précisément le minimum global (ou au moins confirmer la localisation de celui fourni par ce niveau).

3 Second niveau

Le second niveau, initialisé à partir des résultats du premier, est conçu autour de l'emploi de la stratégie multiparamétrique et d'un algorithme d'optimisation locale de type SQP (cf. partie 2.3.3 page 25). Comme nous l'avons vu dans la partie précédente, le minimum fourni par l'algorithme d'optimisation basé sur le métamodèle peut nécessiter d'être affiné, ce qui est l'objectif de ce second niveau. Choisi en raison de ses performances ainsi que de sa capacité à s'adapter à des problèmes avec et sans contraintes, l'algorithme SQP est de plus disponible au sein de la routine `fmincon` de MATLAB.

4 Application de la procédure d'optimisation à des assemblages

On propose dans cette partie de mettre en œuvre la stratégie d'optimisation proposée précédemment sur des cas test d'assemblages. Pour cela, des cas-tests en dimensions 2 et 3 seront introduits. Ils présentent des problématiques du type de celles rencontrées dans la pratique lors du dimensionnement d'un système mécanique.

4.1 Etude de l'impact d'un rainurage sur une répartition de pression

L'exemple d'assemblage présenté ici a pour but d'étudier l'impact d'un rainurage présent entre deux pièces en contact. Dans un contexte de dimensionnement, l'approche consisterait à chercher les dimensions de la rainure permettant d'obtenir une bonne répartition de la pression à l'interface.

La géométrie éclatée du cas-test est proposée en figure 4.9.

Cet exemple est constitué de 15 sous-domaines, comporte $15\,836$ éléments TRI6 et $57\,036$ degrés de liberté. A l'exception de l'interface $\Omega_{10} - \Omega_{11}$ qui comprend du contact et du frottement, toutes les interfaces sont parfaites. Le matériau est supposé ayant un comportement linéaire élastique isotrope avec un module d'Young de $210\,\text{GPa}$ et un coefficient de Poisson de 0.3. On considère une résolution quasi-statique en 5 pas de temps. Un chargement dissymétrique est appliqué linéairement jusqu'à atteindre au dernier pas de temps une valeur de $300\,\text{MPa}$ pour la pression P_1 appliquée sur la sous-structure Ω_{12} et $150\,\text{MPa}$ pour la pression P_2 appliquée sur la sous-structure Ω_{15}.

À l'interface $\Omega_{10} - \Omega_{11}$, un champ de jeu est appliqué. Il modélise la présence d'une rainure dont les dimensions sont données par la figure 4.10. La géométrie de la rainure est pilotée par 3 paramètres : la position X de la rainure par rapport au flanc gauche de

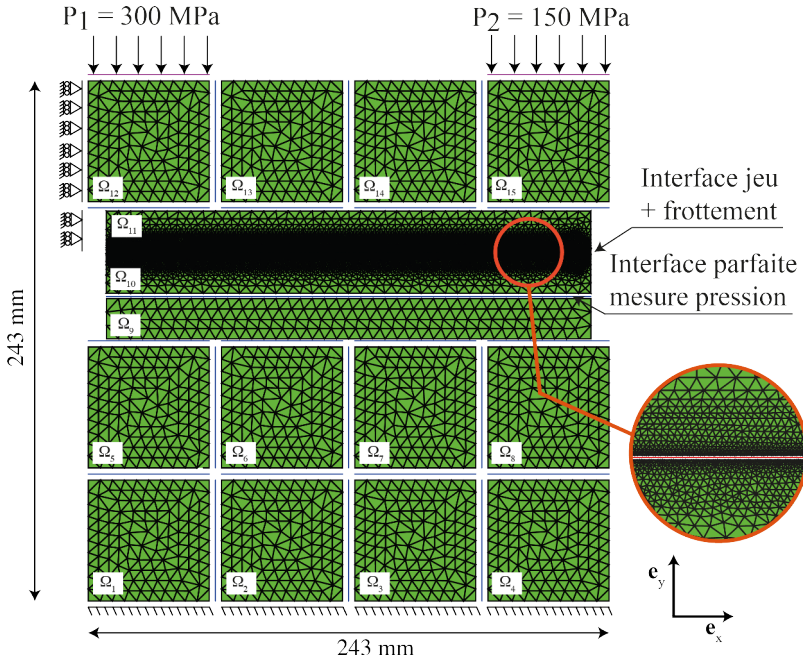


FIGURE 4.9 – Géométrie éclatée du cas-test

la sous-structure, sa hauteur h et sa longueur L. Afin d'éviter de fortes discontinuités du champ de pression à l'interface, la géométrie est construite en employant un polynôme d'ordre 5 et des contraintes de dérivées nulles (du champ de jeu par rapport à e_x) sur les extrémités et au milieu de la rainure. Le choix de ces paramètres est fait judicieusement (en particulier pour X et L), de manière à ce que la rainure ne soit pas débouchante sur les bords de la sous-structure.

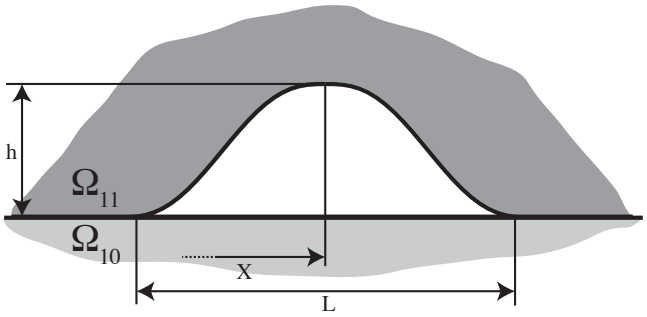


FIGURE 4.10 – Géométrie de la rainure et paramètres

Au niveau de l'interface $\Omega_9 - \Omega_{10}$, le champ de pression est mesuré. Ce dernier est directement impacté par la géométrie de la rainure. On trouvera en figure 4.11 le champ de pression relevé à cette interface dans le cas où la rainure est située en $X = 72.9\,\mathrm{mm}$, a une largeur $L = 24.3\,\mathrm{mm}$ et sa hauteur est fixée à $h = 0.5\,\mathrm{mm}$.

La figure 4.12 présente, quant à elle, la valeur de la composante σ_{yy} dans la structure

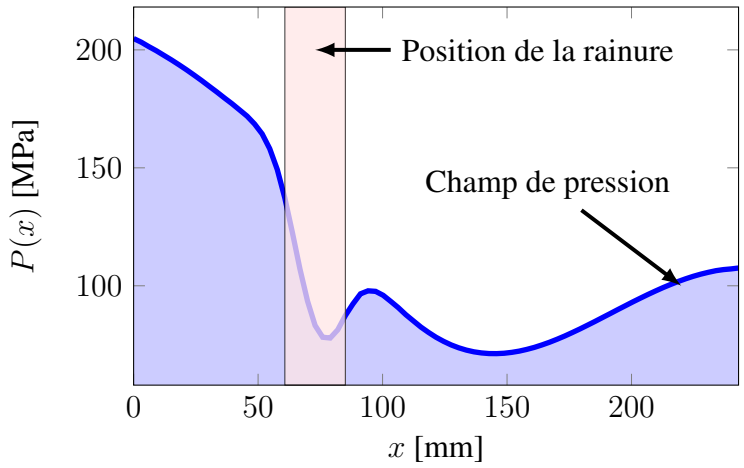


FIGURE 4.11 – Exemple de champ de pression à l'interface $\Omega_9 - \Omega_{10}$ pour $X = 72.9\,\text{mm}$, $L = 24.3\,\text{mm}$ et $h = 0.5\,\text{mm}$ et position de la rainure

complète au dernier pas de temps et pour la rainure définie précédemment.

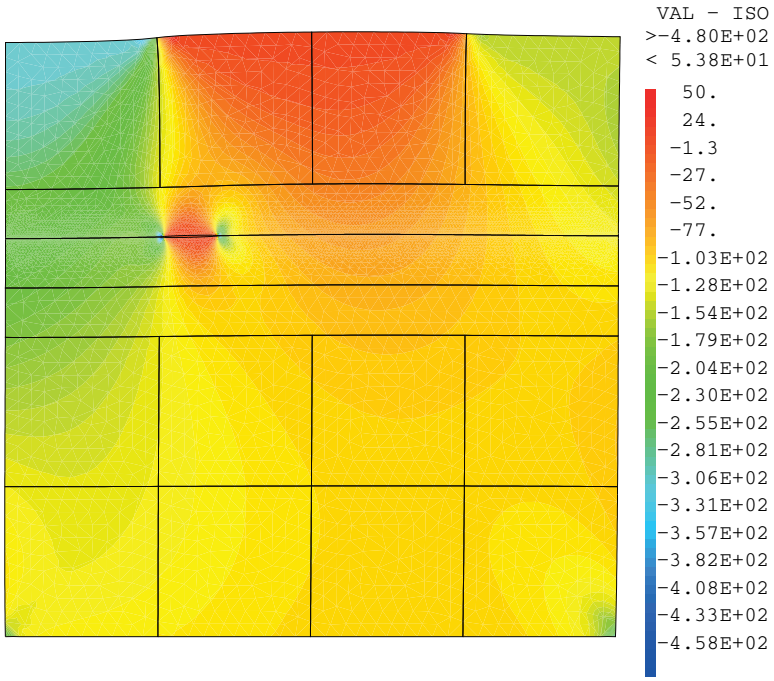


FIGURE 4.12 – Composante σ_{yy} du champ des contraintes

4.1.1 Problème d'optimisation

Suite à la présentation du problème mécanique, nous pouvons envisager le problème d'optimisation. Ce dernier consiste à chercher la meilleure géométrie de la rainure permet-

tant d'assurer une bonne répartition de la pression à l'interface $\Omega_{10} - \Omega_{11}$. C'est-à-dire une répartition la plus homogène possible à l'interface. Pour cela, 4 paramètres de conception peuvent être considérés : les paramètres X, L et h définissant la géométrie de la rainure et le coefficient de frottement μ à l'interface $\Omega_{10} - \Omega_{11}$. Leurs plages sont définies de la manière suivante :

$$\mathbf{x} = (X, L, h, \mu) \in [35\,\text{mm}, 208\,\text{mm}] \times [1\,\text{mm}, 30\,\text{mm}] \times [0\,\text{mm}, 0.5\,\text{mm}] \times [0.05, 0.9]$$

La fonction objectif considérée sera la fonction définie de la manière suivante pour le jeu de paramètres $\mathbf{x}$:

$$f_{\text{obj}} = y(\mathbf{x}) = \frac{P_{\text{max}} - P_{\text{min}}}{P_{\text{moy}}} \tag{4.13}$$

où P_{max}, P_{min} et P_{moy} sont respectivement la pression maximale, la pression minimale et la pression moyenne mesurées à l'interface $\Omega_9 - \Omega_{10}$.

4.1.2 Etude paramétrique

Afin d'illustrer la réponse de l'assemblage aux variations des paramètres, un premier cas comportant 2 paramètres de conception est considéré. Le coefficient de frottement μ et la hauteur de la rainure sont fixés à respectivement 0.6 et 0.25 mm. Les autres coefficients (X et L) sont variables. En considérant un échantillonnage régulier de l'espace des paramètres, on peut alors construire la surface de réponse de la figure 4.13 obtenue en une grille régulière 25×25. Les performances de la stratégie multiparamétrique associée à ces calculs sont regroupées dans le tableau 4.3.

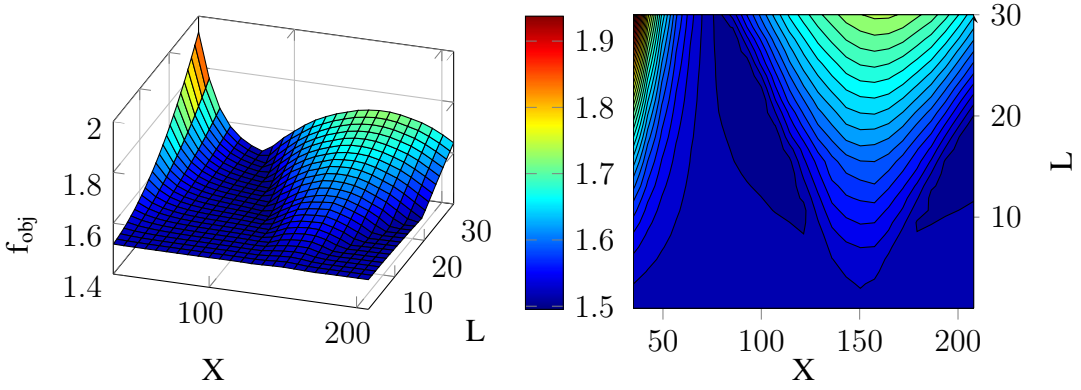


FIGURE 4.13 – Surface de réponse de la fonction objectif considérée obtenue à $\eta = 10^{-4}$

Le ou les minima de cette fonction sont relativement difficiles à obtenir *a priori*. Une étude plus fine montre cependant qu'ils se situent dans une zone définie telle que $(X, L) \in [89\,\text{mm}, 100\,\text{mm}] \times [22\,\text{mm}, 25\,\text{mm}]$. On considèrera le cas d'une optimisation à deux paramètres basée sur la même prise en compte de paramètres que l'étude paramétrique présentée précédemment. On emploie une approche sans gradients et une avec gradients basées sur des métamodèles de type krigeage et cokrigeage.

Grandeurs	Valeurs
Nb. d'appels au solveur	625
Temps CPU total [s]	21 488
Temps CPU moyen [s]	34.4
Temps CPU mini. [s]	4.3
Temps CPU maxi. [s]	208.8
Nb. itérations total.	44 816
Nb. itérations moyen	72
Nb. itérations mini.	10
Nb. itérations maxi.	326
Gain_I	3.90
Gain_M	6.07

TABLE 4.3 – Caractéristiques du calcul mécanique permettant d'obtenir la surface de réponse figure 4.13 pour une grille 25×25

4.1.3 Approche sans enrichissement

On propose tout d'abord de s'intéresser à un processus d'optimisation ne comportant pas de phase d'enrichissement. Le processus consiste simplement à construire un métamodèle, à réaliser une optimisation globale sur ce dernier puis à réutiliser cette information pour initialiser l'algorithme de recherche précise. On va montrer ici que ce processus n'est pas viable dans un cas pratique.

Tout d'abord, les métamodèles de krigeage et cokrigeage sont construits avec un tirage comportant 8 points. Le tableau 4.4 présente les résultats obtenus lors du déroulement du processus.

Au vu de ces résultats, on constate que l'approximation fournie par le métamodèle ne permet pas de fournir un minimum approché $\mathbf{x}^*_{\text{meta}}$ de qualité suffisante. Par ailleurs, la fonction objectif présentant de grandes zones constantes, l'algorithme SQP ne parvient pas à fournir un minimum de meilleure qualité.

Le cas présenté ci-dessus est un cas pathologique pour plusieurs raisons. Le nombre de points pris en compte s'avère, en particulier, trop faible pour construire l'approximation. Deux solutions sont alors envisageables : augmenter le nombre de points initiaux ou ajouter une phase d'enrichissement. On propose de s'intéresser à cette deuxième approche dans la suite.

4.1.4 Approche avec enrichissement

On propose désormais d'intégrer une phase d'enrichissement à la suite de la construction du métamodèle initial. Celle-ci sera intégrée en réalité au sein de l'algorithme EGO dont la convergence sera contrôlée par le critère issu de [HUANG et al. 2006] pour une valeur fixée à 10^{-3}. On considère pour les deux approches des tirages initiaux à 20 points obtenus par IHS.

			Sans gradient	Avec Gradients
Niveau 1	Métamodèle	n_s	8	8
		Nb. appels COFAST	8	24
		Temps CPU [s]	470	875
		$\tilde{y}^*_{\text{meta}}$	1.497	1.487
		$\mathbf{x}^*_{\text{meta}}$	(66.1 mm, 17.5 mm)	(92.6 mm, 1 mm)
Niveau 2	SQP	Nb. ite. SQP	22	1
		Nb. appels COFAST	105	57
		Temps CPU [s]	5451	877
		y^*	1.519	1.515
		$\mathbf{x}^*$	(66.1 mm, 16.1 mm)	(92.0 mm, 1 mm)
Total		Temps CPU[s]	5451	877
		Gain	3.13	8.63

TABLE 4.4 – Mise en défaut de l'approche sans enrichissement sur le cas-test de la rainure

Le tableau 4.5 regroupe les résultats obtenus par des approches avec et sans gradients construites avec 8 points initiaux ainsi qu'un cas de krigeage (sans gradient) construit avec 20 points initiaux.

			Sans gradient		Avec Gradients
Niveau 1	Métamodèle	n_s	8	20	8
		Nb. appels COFAST	8	20	24
		Temps CPU[s]	470	1920	**750**
		$\tilde{y}^*_{\text{meta}}$	1.497	1.478	1.486
		$\mathbf{x}^*_{\text{meta}}$	(66.1 mm, 17.5 mm)	(94.8 mm, 22.7 mm)	(92.6 mm, 1 mm)
	Enrichissement	Nb. ite. EGO	40	34	11
		Nb. appels COFAST	40	34	33
		Temps CPU [s]	1800	2500	**705**
		$\tilde{y}^*_{\text{EGO}}$	1.501	1.476	1.502
		$\mathbf{x}^*_{\text{EGO}}$	(207.1 mm, 13.3 mm)	(89.8 mm, 22.9 mm)	(92.3 mm, 19.6 mm)
Niveau 2	SQP	Nb. ite. SQP	14	3	1
		Nb. appels COFAST	79	12	19
		Temps CPU [s]	3543	450	592
		y^*	1.50	1.490	1.501
		$\mathbf{x}^*$	(207 mm, 13.1 mm)	(89.9 mm, 22.9 mm)	(92.3 mm, 19.56 mm)
Total		Temps CPU[s]	5340	4542	**2047**
		Gain	3.89	3.7	**4.31**

TABLE 4.5 – Résultats du processus d'enrichissement sur le cas-test de la rainure

De ces résultats, on peut déduire un certain nombre de conclusions :

- le métamodèle à 8 points et sans gradients ne permet pas dans ce cas de parvenir à localiser le minimum global. Il converge vers un minimum local dont la valeur est extrêmement proche de celle du véritable minimum global.
- l'augmentation du nombre de points initiaux pris en compte permet à l'approche sans gradient de parvenir à localiser plus précisément le minimum global (avec 8 points, elle converge vers un minimum local).
- lors de l'emploi de l'approche à gradients avec seulement 8 points, l'enrichissement joue un véritable rôle correctif. Le jeu de paramètres optimal passe, en effet, de $(92.6\,\text{mm}, 1\,\text{mm})$ à l'issue de la construction du métamodèle initial à $(92.3\,\text{mm}, 19.6\,\text{mm})$ suite à la phase d'enrichissement.
- On constate que globalement, sur cet exemple, le second niveau n'apporte pas de réelle amélioration du point de vue de la localisation du minimum global. Dans le cas du krigeage à 8 points, celui-ci a plutôt du mal à converger et engendre un surcoût en termes de temps de calcul.

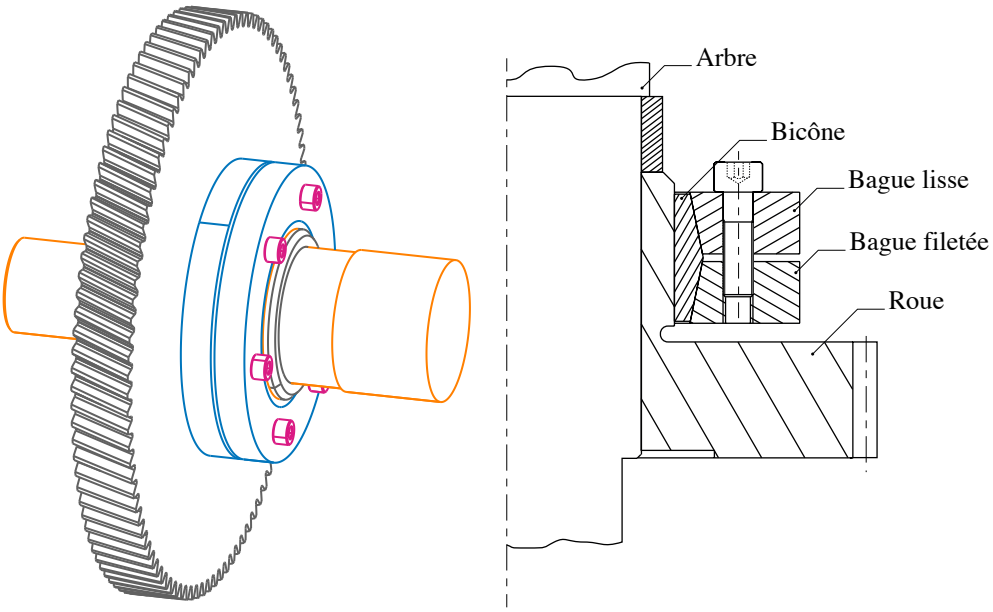
Abstraction faite de la phase d'optimisation fine du second niveau, l'approche basée sur l'usage des gradients et le cokrigeage permet l'obtention du minimum global en $1454\,\text{s}$ (comparés aux $4420\,\text{s}$ de l'approche basée sur le krigeage sans gradient).

Remarque 4.1 : On rappelle que les valeurs $\tilde{y}^*_{\text{meta}}$ et $\tilde{y}^*_{\text{EGO}}$ sont des valeurs du minimum fournies par un métamodèle, ce qui explique le fait qu'ils peuvent avoir des valeurs inférieures à la valeur atteinte en fin de processus.

4.2 Montage fretté

Dans un second temps, un montage fretté est considéré. L'objectif du dispositif est d'assurer un montage par adhérence d'une roue sur un arbre lisse. Une série de vis assure une mise en charge du serrage par l'intermédiaire de bague cônique et d'une bague bicônique (voir figure 4.14).

Le montage considéré ici ne comporte pas d'obstacle à la rotation (de l'arbre par rapport à la roue) et cette liaison repose intégralement sur un phénomène d'adhérence. Un exemple basé sur ce principe sera proposé par la suite. L'objectif est de parvenir à dimensionner les grandeurs intervenant dans la réalisation de la liaison pour que celle-ci parvienne à transmettre un couple maximum. L'approche consiste à considérer une géométrie axisymétrique. Celle-ci, présentée en figure 4.15, correspond à un demi-montage fretté où une seule bague est prise en compte. Le modèle proposé comporte 4 sous-structures et 3 interfaces. Le maillage considéré compte 3800 éléments QUA4. On considère que l'ensemble des pièces est en acier (Module de Young de $210\,\text{GPa}$ et coefficient de Poisson de 0.3). Le chargement de $22\,\text{MPa}$ est appliqué de manière quasi-statique sur la bague (en 10 pas de temps). Ce dernier est calculé en tenant compte du couple de serrage (donnée constructeur) des 10 vis que compte la frette, du frottement sous la tête de vis et au niveau des filets.



(a) Vue 3D **(b)** Mise en plan

FIGURE 4.14 – Exemple d'assemblage démontable par frette conique

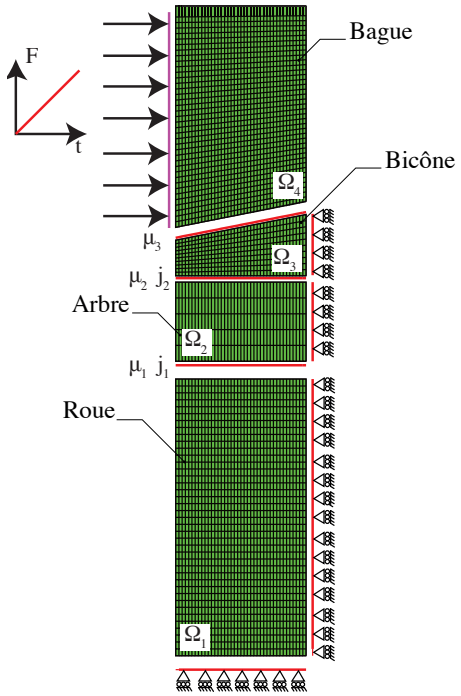


FIGURE 4.15 – Problème axi-symétrique 2D

Le cas-test d'optimisation comporte 5 variables de conception :

- dont 3 coefficients de frottement
  - μ_1 à l'interface $\Omega_1 - \Omega_2$ entre l'arbre et la roue ($0.05 \leq \mu_1 \leq 0.5$) ;
  - μ_2 à l'interface $\Omega_2 - \Omega_3$ entre la roue et le bicône ($0.05 \leq \mu_2 \leq 0.5$) ;
  - μ_3 à l'interface $\Omega_3 - \Omega_4$ entre le bicône et la bague ($0.05 \leq \mu_3 \leq 0.5$).
- et 2 jeux :
  - j_1 à l'interface $\Omega_1 - \Omega_2$ entre l'arbre et la roue $0\,\mu\text{m} \leq j_1 \leq 49\,\mu\text{m}$) ;
  - j_2 à l'interface $\Omega_2 - \Omega_3$ entre la roue et le bicône $36\,\mu\text{m} \leq j_2 \leq 106\,\mu\text{m}$).

La fonction objectif considérée sera, comme annoncé précédemment, le couple maximum transmissible que l'on cherche à maximiser. Ce dernier est calculé à partir du coefficient de frottement à l'interface $\Omega_1 - \Omega_2$ et de l'effort normal transitant dans cette interface.

En considérant pour le krigeage et le cokrigeage, des tirages initiaux comportant 20 points échantillonnés par IHS, on constate dans le tableau 4.6 que le premier niveau de l'approche parvient à localiser quasiment directement le minimum global de la fonction. Les métamodèles initiaux sont construits en 20 points échantillonnés par IHS (soit 20 appels au solveur pour le krigeage et 120 pour le cokrigeage).

			Sans gradient		Avec Gradients	
Niveau 1	Métamodèle	n_s	10	20	10	20
		Nb. appels COFAST	10	20	60	120
		Temps CPU[s]	137	244	267	525
		y^*_{meta}	-1.29×10^4 Nm	-1.66×10^4 Nm	-1.70×10^4 Nm	-1.982×10^4 Nm
		$\mathbf{x}^*_{\text{meta}}$	$(0.5, 0.5, 0.05, 0\,\mu\text{m}, 36\,\mu\text{m})$			
	Enrichissement	Nb. ite. EGO	1	0	1	1
		Nb. appels COFAST	1	0	6	6
		Temps CPU [s]	27.4	0	28.7	28.2
		y^*_{EGO}	-2.16×10^4 Nm	y^*_{meta}	-2.16×10^4 Nm	
		$\mathbf{x}^*_{\text{EGO}}$	$(0.5, 0.5, 0.05, 0\,\mu\text{m}, 36\,\mu\text{m})$			
Niveau 2	SQP	Nb. ite. SQP	1	1	1	5
		Nb. appels COFAST	7	7	7	68
		Temps CPU [s]	87.3	92.7	87.4	267
		y^*	-2.16×10^4 Nm			
		$\mathbf{x}^*$	$(0.5, 0.49, 0.05, 0\,\mu\text{m}, 36\,\mu\text{m})$			$(0.5, 0.44, 0.05, 0\,\mu\text{m}, 36\,\mu\text{m})$
Total		Temps CPU[s]	**238**	336	383	820
		Gain	1.68	1.32	**4.54**	4.1

TABLE 4.6 – Résultats du processus d'optimisation sur la frette comportant 5 paramètres

Le cas-test de cette frette axi-symétrique a la particularité de présenter une fonction objectif facile à approximer. En effet, dans ce cas, dès la fin de la construction du métamodèle initial, le minimum est localisé. On constate cependant que

- dans le cas du krigeage à 20 poins initiaux, l'algorithme EGO ne procède à aucune itération. Le critère d'arrêt est en effet vérifié sur le métamodèle initial. Il nécessite cependant l'emploi de l'optimisation précise du second niveau pour obtenir la bonne valeur de la fonction objectif au minimum.
- dans les autres cas, une seule itération de l'EGO permet sa convergence. Un nouveau point est ajouté et ce dernier permet alors d'obtenir la bonne valeur du minimum.
- dans le cas du cokrigeage à 20 points initiaux, on observe un changement de valeur du paramètre μ_2 lors de la phase d'optimisation locale qui a en réalité très peu d'impact sur le couple transmissible. On se trouve en présence d'une « vallée » de minima où l'algorithme SQP peine à converger.

Dans ce cas, l'approche par krigeage est clairement plus performante. Cela s'explique par la grande simplicité de la fonction objectif, qui ne nécessiterait pas l'emploi d'un métamodèle aussi riche que le cokrigeage mais dont l'élaboration est un peu plus chère.

4.3 Joint d'accouplement

Dans cette partie, un cas-test qui à terme serait traité avec l'approche proposée est introduit. On ne s'intéressera ici qu'à introduire les performances de la stratégie multiparamétrique sur cet exemple.

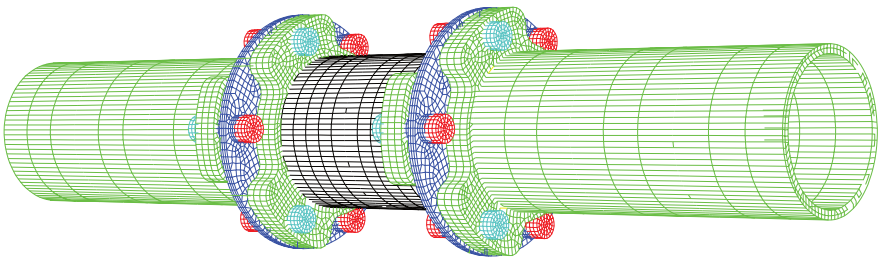
Issu d'une adaptation libre d'un joint d'accouplement de type ROBA-DS de la marque Mayr [1], cet exemple consiste en un joint d'accouplement. Il permet l'accouplement sans jeu de deux arbres tout en tolérant des défauts mineurs d'alignement. La modélisation retenue est proposée dans la figure 4.16.

L'assemblage comporte ici 50 sous-structures, 89 interfaces et 144588 degrés de liberté. Toutes les pièces de l'assemblage seront supposées en acier (module d'Young de 210 GPa et coefficient de Poisson de 0.3). Le chargement mécanique quasi-statique sera imposé en 3 temps :

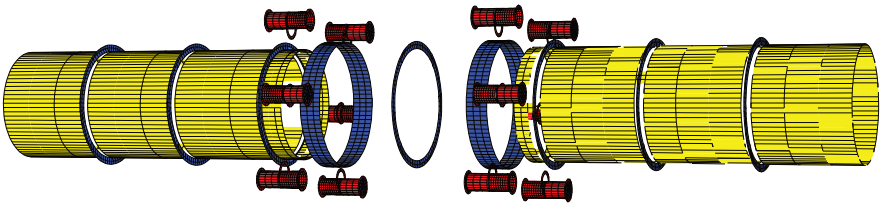
1. la précharge de l'ensemble des 12 vis est réalisée en un pas de temps.
2. la mise en flexion du joint en 5 pas de temps. Un angle de $1.4°$ est imposé entre l'entrée et la sortie. Cette phase simule un désalignement des deux arbres accouplés par le dispositif.
3. la sollicitation en torsion de l'ensemble par l'intermédiaire d'un angle de $1.5°$ imposé entre l'entrée et la sortie du joint. Cette phase modélise la mise en charge du joint.

Le rattrapage du désalignement des arbres est réalisé par l'intermédiaire de deux plaques assemblées par l'intermédiaire de 2×3 vis avec l'entrée (ou la sortie) et de 2×3 vis avec la partie centrale du joint d'accouplement. On considérera la présence de frottement entre toutes les pièces de l'assemblage. De cet assemblage, il est possible de

1. `http://www.mayr.com`



(a) Géométrie de l'accouplement



(b) Interfaces

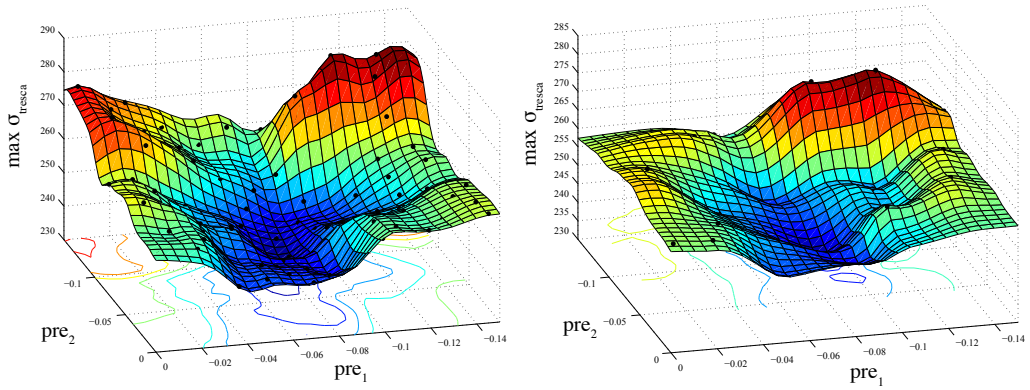
FIGURE 4.16 – Joint d'accouplement

considérer un ensemble de cas-tests prenant en compte divers paramètres : les précharges dans les vis et les coefficients de frottement à toutes les interfaces.

D'un point de vue optimisation, il est possible de chercher à minimiser la contrainte de cisaillement (via le calcul d'une contrainte équivalente de Tresca par exemple) maximale dans l'ensemble des vis pour un ensemble de paramètres de serrage.

Seule une étude de faisabilité a été réalisée. Celle-ci consiste à considérer deux précharges variables pour les deux groupes de 2×3 vis liant la partie centrale et l'entrée et la sortie du joint. On retrouve, en figure 4.17, l'évolution de la contrainte équivalente de Tresca maximale dans la vis la plus chargée en fonction des précharges dans ces vis (les valeurs des précharges sont imposées par l'intermédiaire de jeux dont les valeurs sont comprises entre $0\,\mu m$ et $150\,\mu m$). Les surfaces de réponses proposées dans la figure 4.17 on été obtenues en employant un métamodèle de krigeage ordinaire et de cokrigeage à gradients construits respectivement avec 100 (sans gradients) et 20 points (avec gradients).

L'emploi de la stratégie multiparamétrique permet dans ce cas une réelle réduction des temps de calcul et permet de construire un métamodèle de cokrigeage présentant une très bonne localisation du minimum global. Le métamodèle de krigeage à 100 points a nécessité 9h pour être obtenu (3.5 jours sans la stratégie multiparamétrique) tandis que le métamodèle de cokrigeage a nécessité que 4h pour être construit (30h sans stratégie multiparamétrique). La réalisation d'une optimisation globale basée sur l'EGO pourrait ensuite être mise en œuvre.



(a) Approximation par krigeage ($n_s = 100$) **(b)** Approximation par cokrigeage à gradients ($n_s = 20$)

FIGURE 4.17 – Contrainte de cisaillement maxi dans la vis la plus chargée de l'assemblage

5 Bilan de la stratégie d'optimisation proposée

Ce chapitre présente la mise en œuvre d'une démarche d'optimisation à deux niveaux basée sur les différents outils introduits dans les chapitres précédents. Ainsi, le premier niveau s'articule autour d'une optimisation globale réalisée sur un métamodèle de type krigeage ou cokrigeage. Celui-ci est construit en utilisant les réponses et les gradients fournis par la stratégie multiparamétrique. Le second niveau repose, quant à lui, sur une optimisation locale et précise assurant l'obtention de l'optimum global en utilisant la stratégie multiparamétrique. Au sein du premier niveau, une procédure d'enrichissement classiquement employée sur les modèles de krigeage est rappelée et utilisée. Dans un dernier temps, quelques exemples d'assemblages ont été présentés et permettent d'illustrer les performances de la méthode. On s'intéresse à la fois à la précision de la détermination du minimum et aux performances en termes de temps de calcul.

La stratégie d'optimisation proposée présente une réelle capacité à résoudre des problèmes d'optimisation globale sans contrainte portant sur des assemblages. Les performances en termes de localisation du minimum sont obtenues en grande partie par l'usage du processus d'enrichissement basé sur l'amélioration espérée. En effet, cette phase apparaît comme cruciale et assure des ajouts (si nécessaire) de points pertinents conduisant à une amélioration de la localisation et de la valeur du minimum que fournit le métamodèle en fin du déroulement du premier niveau. On constate par ailleurs que sur les exemples présentés, le second niveau n'engendre pas de réelle amélioration de la qualité du minimum. Il nécessite un nombre non négligeable de calculs, ce qui augmente par conséquent le temps de calcul nécessaire à l'optimisation. Dans le cas de fonctions objectifs plus multimodales ou pour des exemples comportant plus de variables, l'optimisation locale du niveau 2 pourrait avoir une action importante et nécessiterait une étude complémentaire.

Le calcul des gradients et leur utilisation pour construire le modèle de substitution permet une réduction des temps de calcul tout en permettant l'obtention d'un minimum de

qualité. Globalement, dans le cadre des problèmes traités ici, on observe que les fonctions objectifs considérées sont souvent peu non-linéaires et peu multimodales. Les métamodèles nécessitent alors relativement peu de points. C'est pour cette raison que la règle proposée par [D. R. JONES, SCHONLAU et WELCH 1998] consistant à considérer $10n_p$ points (n_p est le nombre de paramètres du problème) pour construire le métamodèle initial n'a pas été retenue ici (il a été décidé de se contenter de $4n_p$ points). Ce choix se justifie également par la qualité d'approximation que peut fournir le cokrigeage avec des réponses et gradients connus en un nombre plus faible de points.

Conclusions et perspectives

Les travaux proposés dans cette thèse visent à élaborer une méthode dédiée à l'optimisation d'assemblages. L'obtention de solutions d'un problème mécanique d'assemblage requiert de coûteuses résolutions d'un problème fortement non-linéaire. Ce constat montre *a priori* l'incompatibilité de l'optimisation globale, qui nécessite de très nombreuses évaluations de la fonction objectif pour converger, et du calcul mécanique d'assemblage. La méthode proposée se positionne alors comme une alternative à une optimisation directe sur le modèle éléments finis. L'idée consiste à s'appuyer sur une méthode d'**optimisation multiniveaux de modèle** basée sur deux éléments clés pour réduire les temps de calcul :

- une méthode de résolution mécanique capable de résoudre de manière robuste un ensemble de problèmes d'assemblage similaires ayant chacun quelques paramètres (coefficients de frottement, précharges, jeux...) qui diffèrent. Cette première problématique est résolue grâce à l'emploi de la **stratégie multiparamétrique** basée sur la méthode LATIN.

- un modèle de substitution capable de fournir des réponses approchées de la fonction objectif considérée en un temps de calcul très faible. Les métamodèles employés dans la méthode proposée sont principalement de type krigeage et **cokrigeage à gradients**. Ils permettent alors d'obtenir un minimum approché solution du problème d'optimisation globale.

Ces travaux s'inscrivent dans une démarche faisant intervenir 3 grandes thématiques : celle du **calcul d'assemblages**, celle des **modèles de substitution** et celle de l'**optimisation**. Ces 3 grands axes ont fait l'objet, dans le premier chapitre, d'une étude bibliographique poussée permettant de fixer un cadre clair pour l'optimisation d'assemblages.

Tout d'abord, le contexte de l'optimisation est posé par l'intermédiaire d'un certain nombre de rappels concernant la définition des problèmes classiques d'optimisation paramétrique et les méthodes pour les résoudre. Une part importante de cette partie concerne l'optimisation multiniveaux. Dans un second temps, la **méthode LATIN pour les assemblages** est présentée. Elle permet, en se basant sur un principe de séparation des difficultés, de résoudre de manière non incrémentale des problèmes fortement non-linéaires tels que les problèmes d'assemblages. Les spécificités de cette méthode ont conduit à la mise au point de la **stratégie multiparamétrique** qui permet, grâce à une réinitialisation judicieuse de

l'algorithme de la méthode LATIN, de significatives réductions des temps de calcul liés aux multiples résolutions de problèmes d'assemblages similaires. Parmi les nombreuses méthodes existantes, elle s'impose comme un outil particulièrement performant. C'est finalement par un rappel concernant la mise en œuvre des métamodèles que s'achève l'étude bibliographique.

Dans le deuxième chapitre, la stratégie multiparamétrique, qui jusqu'à présent avait été principalement utilisée pour réaliser des études paramétriques, est employée pour **calculer à moindre coût** réponses et gradients d'une fonction objectif pour un ensemble de jeux de paramètres. En s'appuyant sur un ensemble de cas-tests d'assemblages représentatifs des différents phénomènes pouvant survenir avec ce type de problèmes, les différents aspects et paramétrages de la stratégie multiparamétrique sont étudiés dans l'objectif de fournir de manière **robuste** les réponses et **gradients** d'une fonction objectif considérée. La stratégie multiparamétrique apparaît alors comme une méthode clé pour l'optimisation d'assemblages. Elle permet, en effet, de **fortes réductions des temps de calcul**. Elle assure sur certains cas une division par 10 du temps associé à de multiples résolutions de problèmes présentant de fortes non-linéarités comme le frottement ou le contact (en comparaison avec à la méthode LATIN sans stratégie multiparamétrique). Les éléments du deuxième chapitre montrent que la méthode nécessite un certain nombre de paramétrages afin de permettre une réduction maximale des temps de calcul. En particulier, le choix de la stratégie d'initialisation est incontournable. Il a été montré qu'une stratégie d'initialisation basée sur la recherche d'une solution proche dans l'espace des paramètres assure l'obtention des meilleures performances. C'est dans ce contexte que le calcul des gradients peut être mis en œuvre. Ainsi, une méthode de **différences finies** est proposée. Une étude est menée pour en déterminer le pas. Bien que complexe en raison du bruit numérique constaté lors de l'usage de la stratégie multiparamétrique, elle conduit au choix du paramètre. Les réponses nécessaires aux différences finies étant obtenues en des points très proches dans l'espace des paramètres, la stratégie multiparamétrique montre des performances considérables (typiquement entre deux points proches les gains peuvent être de l'ordre de 100).

Fort de cette capacité d'obtention des gradients, le troisième chapitre propose de s'intéresser à l'élaboration de **modèles de substitution** construits non seulement à partir de réponses, mais également à partir des gradients de la fonction objectif connus en seulement quelques points. Ainsi, après quelques développements théoriques autour du **krigeage** et des fonctions de base radiales sans gradients, leurs extensions capables de tenir compte des gradients sont détaillées. Leur qualité d'approximation globale est étudiée en détail sur un ensemble de fonctions analytiques plus ou moins non-linéaires et multimodales. Finalement, la stratégie multiparamétrique est employée pour fournir réponses et gradients. La faisabilité de la construction d'un modèle de substitution du modèle mécanique est alors étudiée. La qualité d'approximation est également analysée sur un ensemble d'exemples d'assemblages. La prise en compte de l'information gradients dans un modèle de substitution conduit à des approximations de très bonne qualité. On constate en effet que

- pour un même nombre de points, l'ajout des gradients dans l'approximation accroît très nettement sa qualité.

- il est possible de réduire le nombre de points où l'on connaît les réponses et gradients et de construire un modèle de substitution de qualité équivalente à un métamodèle sans gradient comportant plus de points.

L'étude de qualité réalisée montre la grande robustesse proposée par les métamodèles de type krigeage et **cokrigeage à gradients**. Ce sont ces deux types de métamodèles qui sont employés dans la phase d'optimisation.

La réunion de ces deux outils forme ensuite la base de l'**optimisation à deux niveaux de modèles** proposée dans le quatrième chapitre. Le premier niveau est constitué d'une optimisation globale incluant un procédé d'**enrichissement** « intelligent », tandis que le second consiste en une optimisation locale utilisant directement la stratégie multiparamétrique. Quelques cas-tests sont ensuite employés pour étudier les performances de la méthode. L'optimisation conduite sur les métamodèles intègre un dispositif d'enrichissement basé sur le principe de l'amélioration espérée. Ce dispositif apparaît comme incontournable en permettant l'obtention du minimum global et sa localisation plus précise. Il faut cependant noter que dans le cas du cokrigeage à gradients, l'ajout d'un nouveau point conduit à l'apport d'une information plus riche que dans une méthode sans gradient. Ce processus d'enrichissement, par ailleurs, conduit à la non nécessité d'utiliser le second niveau de l'approche considérée de manière quasi systématique. Il permet cependant un réelle validation du minimum proposé par le premier niveau en utilisant le modèle mécanique. Dans le cas d'une fonction objectif complexe, le cokrigeage s'impose comme la méthode d'approximation permettant la localisation à moindre coût du minimum global.

Globalement, la réalisation d'une optimisation sur un assemblage m'a conduit à élaborer deux outils spécifiques. Le premier forme la toolbox COMET (***CO**ntrol **M**ultiparametric strat**E**gy **T**oolbox*) assurant la communication entre MATLAB et le solveur COFAST3D qui intègre la stratégie multiparamétrique basée sur la méthode LATIN. Le second outil forme la toolbox GRENAT (***GR**adient **EN**hanced **A**pproximation **T**oolbox*) intégrant les métamodèles de krigeage, cokrigeage à gradients, ainsi que les fonctions de base radiales avec et sans gradients. Ces deux outils ont été conçus pour être parfaitement interfaçables et assurer un environnement propice à la conduite de problèmes d'optimisation d'assemblages.

Dans la continuité des travaux proposés, un grand nombre de perspectives sont envisageables :

- à court terme :
  - l'application de la méthode d'optimisation sur des exemples d'assemblages industriels de plus grande taille (nombre de variables et nombre de degrés de liberté) permettrait de confirmer les résultats observés dans ces travaux.
  - pour se rapprocher davantage de problèmes industriels, la prise en compte des fonctions contraintes est incontournable. L'usage d'un métamodèle pour approcher ces nouvelles fonctions est envisageable.

- ▷ le processus d'enrichissement basé sur l'amélioration espérée pourrait laisser place à l'amélioration espérée pondérée renforçant l'étude exploratoire. La faisabilité de cette approche pourrait être étudiée facilement (elle a été implémentée mais non utilisée).

- ► à moyen terme :
  - ▷ la problématique du calcul des gradients mériterait des travaux complémentaires relatifs au choix robuste du pas des différences finies. Pour cela, une étude poussée liant recherche du pas des différences finies et usage de la stratégie multiparamétrique est incontournable. Il serait nécessaire d'étudier en particulier la relation entre le critère d'arrêt de la méthode LATIN, le maillage, les phénomènes non-linéaires et la précision du solveur. Les travaux de [MORÉ et WILD 2011 ; MORÉ et WILD 2012a ; MORÉ et WILD 2012b] pourraient fournir des éléments d'aide.
  - ▷ l'intégration d'indicateurs d'erreurs sur des quantités d'intérêt locales permettrait un contrôle plus fin de la qualité de la solution et de la fonction objectif considérée.
  - ▷ du point de vue des métamodèles, la prise en compte de l'intégralité des gradients peut s'avérer dans certain cas inutile (lorsque deux points sont proches, par exemple). Une solution serait de construire le modèle de substitution avec des données partielles ou manquantes afin de ne tenir compte que de l'information « utile ». Pour cela, on pourra se référer aux travaux de [FORRESTER, SÓBESTER et KEANE 2006].
  - ▷ un autre axe consisterait à tirer partie de la possibilité, grâce à la méthode LATIN, d'obtenir des solutions partiellement convergées pour construire le métamodèle (les premiers travaux dans ce sens sont prometteurs [COURRIER, BOUCARD et SOULIER 2013]).

- ► à plus long terme, l'utilisation de la méthode PGD dans le cadre de l'optimisation devrait permettre une réduction supplémentaire des temps de calcul. Elle nécessite d'importants développements tant la prise en compte des variabilités paramétriques de manière systématique s'avère complexe (voir par exemple [CHINESTA, LEYGUE et al. 2013]).

Bibliographie

ABABOU, R., A. BAGTZOGLOU et E. WOOD (1994). « On the condition number of covariance matrices in kriging, estimation, and simulation of random fields ». In : *Mathematical Geology* 26.1, p. 99–133 (cf. p. 154).

ADELMAN, H. M. et R. T. HAFTKA (1986). « Sensitivity Analysis of Discrete Structural Systems ». In : *AIAA Journal* 24.5, p. 823–832 (cf. p. 125).

ALEXANDROV, N. M., J. E. DENNIS et al. (1998). « A trust-region framework for managing the use of approximation models in optimization ». In : *Structural and Multidisciplinary Optimization* 15.1, p. 16–23 (cf. p. 65).

ALEXANDROV, N. M. et R. M. LEWIS (2000). « Analytical and computational aspects of collaborative optimization ». In : *NASA Technical Memorandum 210104* (cf. p. 37).

ALLAIRE, G. (2007). *Conception optimale de structures*. T. 58. Mathématiques & Applications. Springer-Verlag, Berlin (cf. p. 12).

ALLIX, O. (1992). « Damage analysis of delamination around a hole ». In : *New Advances in Computational Structural Mechanics,* Ladevèze & Zienkiewicz (Ed.) Elsevier Science Publishers B.V., p. 411–421 (cf. p. 41).

ALLIX, O. et P. VIDAL (2002). « A new multi-solution approach suitable for structural identification problems ». In : *Computer Methods in Applied Mechanics and Engineering* 191.25–26, p. 2727–2758 (cf. p. 41).

AMMAR, A., F. CHINESTA et al. (2012). « Proper generalized decomposition of time-multiscale models ». In : *International Journal for Numerical Methods in Engineering* 90.5, p. 569–596 (cf. p. 58).

AMMAR, A. et al. (2007). « A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids : Part II : Transient simulation using space-time separated representations ». In : *Journal of Non-Newtonian Fluid Mechanics* 144.2-3, p. 98–121 (cf. p. 57).

AMMAR, A. et al. (2006). « A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids ». In : *Journal of Non-Newtonian Fluid Mechanics* 139.3, p. 153–176 (cf. p. 57).

ANDERSON, R. L. (1953). « Recent Advances in Finding Best Operating Conditions ». In : *Journal of the American Statistical Association* 48.264, pages (cf. p. 26).

ARNAUD, M. et X. EMERY (2000). *Estimation et interpolation spatiale*. Hermes Science Publications, Paris (cf. p. 161).

ARROW, K., L. HURWICZ et H. UZAWA (1958). *Studies in Nonlinear Programming*. Stanford, CA : Stanford University Press (cf. p. 24).

ARZT, M. (1994). « Approche des phénoménes cycliques par la méthode à grand incrément de temps ». Thèse de doct. école Normale Supérieure de Cachan, 160 P. (Cf. p. 41).

BACK, T., U. HAMMEL et H.-P. SCHWEFEL (1997). « Evolutionary computation : comments on the history and current state ». In : *Evolutionary Computation, IEEE Transactions on* 1.1, p. 3–17 (cf. p. 28).

BAILLARGEON, S. (2005). *Le krigeage : revue de la théorie et application à l'interpolation spatiale de données de précipitations*. Université Laval (cf. p. 136).

BARTHELEMY, B., R. T. HAFTKA et G. A. COHEN (1989). « Physically based sensitivity derivatives for structural analysis programs ». In : *Computational Mechanics* 4.6, p. 465–476 (cf. p. 125).

BARTON, R. R. (1990). *Experiments in Computing Finite Difference Derivatives when Optimizing Low Accuracy Functions*. Rapp. tech. 884. School of Operations Research et Industrial Engineering, Cornell University (cf. p. 123).

BATES, R. A. et al. (1996). « Experimental Design and Observation for Large Systems ». In : *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1, pages (cf. p. 67).

BEACHKOFSKI, B. et R. GRANDHI (2002). « Improved Distributed Hypercube Sampling ». In : *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. AIAA 2002–1274 (cf. p. 70).

BECKER, R. W. et G. LAGO (1970). « A global optimization algorithm ». In : *Proceedings of the 8th Allerton Conference on Circuits and Systems Theory*, p. 3–12 (cf. p. 27).

BEDA, L. et al. (1959). « Programs for automatic differentiation for the machine BESM ». In : *Inst. Precise Mechanics and Computation Techniques, Academy of Science, Moscow* (cf. p. 126).

BENDSØE, M. P. (1995). *Optimization of structural topology, shape, and material*. Springer (cf. p. 13, 36).

BENDSØE, M. P. et O. SIGMUND (2003). *Topology optimization : theory, methods and applications*. Springer-Verlag (cf. p. 13).

BEYLKIN, G. et M. MOHLENKAMP (2005). « Algorithms for Numerical Analysis in High Dimensions ». In : *SIAM Journal on Scientific Computing* 26.6, p. 2133–2159 (cf. p. 57).

BISCHOF, C. et al. (1992). « ADIFOR-Generating Derivative Codes from Fortran Programs ». In : *Sci. Program.* 1.1, p. 11–29 (cf. p. 127).

BISHOP, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press (cf. p. 82).

BLANZÉ, C., L. CHAMPANEY et P.-A. BOUCARD (2004). « Advanced tools for design and analysis for assemblies of structures with uncertainties ». In : *ECCOMAS 2004 - 4th European Congress on Computational Methods in Applied Sciences and Engineering* (cf. p. 59).

BLANZÉ, C., L. CHAMPANEY, J.-Y. COGNARD et al. (1996). « A modular approach to structure assembly computations : application to contact problems ». In : *Engineering Computations* 13.1, p. 15–32 (cf. p. 40, 50).

BLANZÉ, C., L. CHAMPANEY et P. VEDRINE (2000). « Contact problems in the design of a superconducting quadrupole prototype ». In : *Engineering Computations* 17.2, p. 136–153 (cf. p. 41).

BLANZÉ, C., R. DANWÉ et al. (1992). « A new simplified method for the analysis of 2D structures ». In : *28th International Congress of Theorical & Applied Mechanics, Haïfa, Israël* (cf. p. 40).

BOENDER, C. et al. (1982). « A stochastic method for global optimization ». In : *Mathematical Programming* 22.1, p. 125–140 (cf. p. 26).

BOGGS, P. T. et J. W. TOLLE (1995). « Sequential quadratic programming ». In : *Acta numerica* 4.1, p. 1–51 (cf. p. 25).

BOISSE, P. (1987). « Nouvel algorithme à grand incrément de temps pour le calcul des structures élastoplastiques ». Thèse de doct. Université Pierre et Marie Curie, 1 vol. (130 p.) (Cf. p. 41).

BOMPARD, M. (2011). « Modéles de substitution pour l'optimisation globale de forme en aérodynamique et méthode locale sans paramétrisation ». Thèse de doct. Université de Nice Sophia Antipolis (cf. p. 83, 85, 151, 173).

BONABEAU, E., M. DORIGO et G. THERAULAZ (1999). *Swarm intelligence : from natural to artificial systems*. Oxford university press, New York, NY (cf. p. 28).

BOOLE, G. et J. F. MOULTON (1880). *A treatise on the calculus of finite differences*. Macmillan (cf. p. 120).

BOUCARD, P.-A. (1996). « Approche à grand incrément de tempts en grandes transformations ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41).

— (2001). « Application of the LATIN method to the calculation of response surfaces ». In : *Proceeding of the First MIT Conference on Computational Fluid and Solid Mechanics, Cambridge, USA*. T. 1, p. 78–81 (cf. p. 5, 41, 59).

BOUCARD, P.-A., S. BUYTET et P.-A. GUIDAULT (2007). « Une stratégie multi-échelle pour l'étude paramétrique de détails géométriques au sein de structures en contacts multiples ». In : *Revue Européenne des éléments Finis* 16.8, p. 1011–1036 (cf. p. 41).

— (2009). « A multiscale strategy for structural optimization ». In : *International Journal for Numerical Methods in Engineering* 78.1, p. 101–126 (cf. p. 60).

BOUCARD, P.-A., S. BUYTET, B. SOULIER et al. (2010). « Multilevel Modeling ». In : *Multidisciplinary Design Optimization in Computational Mechanics*. Sous la dir. de P. BREITKOPF et R. F. COELHO. John Wiley & Sons, Inc., p. 199–263 (cf. p. 33–34).

BOUCARD, P.-A. et L. CHAMPANEY (2003). « A suitable computational strategy for the parametric analysis of problems with multiple contact ». In : *International Journal for Numerical Methods in Engineering* 57.9, p. 1259–1281 (cf. p. 5, 59).

— (2004). « Approche multirésolution pour l'étude paramétrique d'assemblages par contact et frottement ». In : *Revue Européenne des éléments* 13.5-7, p. 437–448 (cf. p. 59).

BOUCARD, P.-A. et P. LADEVÈZE (1999). « A multiple solution method for non-linear structural mechanics ». In : *Mechanical Engineering* 50.5, p. 317–328 (cf. p. 5, 41, 59).

BOUCARD, P.-A., D. ODIÉVRE et F. GATUINGT (2011). « A parallel and multiscale strategy for the parametric study of transient dynamic problems with friction ». In : *International Journal for Numerical Methods in Engineering* (cf. p. 60).

BOX, G. E. P. (1957). « Evolutionary Operation : A Method for Increasing Industrial Productivity ». In : *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 6.2, pages (cf. p. 20).

BOX, G. E. P. et N. R. DRAPER (1959). « A Basis for the Selection of a Response Surface Design ». In : *Journal of the American Statistical Association* 54.287, p. 622–654 (cf. p. 75).

— (1987). *Empirical model-building and response surfaces*. John Wiley & Sons, New York (cf. p. 77).

BOX, G. E. P., J. S. HUNTER et W. G. HUNTER (1978). « Statistics for experimenters : an introduction to design, data analysis, and model building ». In : *Wiley series in probability and mathematical statistics.* (Cf. p. 66).

BOX, G. E. P. et K. WILSON (1951). « On the Experimental Attainment of Optimum Conditions ». In : *Journal of the Royal Statistical Society. Series B (Methodological)* 13.1, p. 1–45 (cf. p. 63, 77).

BREITKOPF, P. et al. (2005). « Moving least squares response surface approximation : Formulation and metal forming applications ». In : *Computers & Structures* 83.17–18. Advances in Meshfree Methods, p. 1411–1428 (cf. p. 79).

BRÉZILLON, P. et al. (1981). « Numerical estimation of the first order derivative : approximate evaluation of an optimal step ». In : *Computers & Mathematics with Applications* 7.4, p. 333–347 (cf. p. 123).

BROOMHEAD, D. et D. LOWE (1988). « Multivariable functional interpolation and adaptive networks ». In : *Complex Systems* 2, p. 321–355 (cf. p. 77, 156).

BROYDEN, C. G. (1970). « The convergence of a class of double-rank minimization algorithms ». In : *IMA Journal of Applied Mathematics* 6.1, p. 76–90 (cf. p. 22).

BUYTET, S., P.-A. BOUCARD et P.-A. GUIDAULT (2008). *Optimisation multi-niveaux de modéles – Application à l'optimisation des paramétres de liaison dans les assemblages*. Rapp. tech. ANR-05-RNTL-009. ANR Projet OMD (cf. p. 60).

BYRD, R., R. SCHNABEL et G. SHULTZ (1987). « A Trust Region Algorithm for Nonlinearly Constrained Optimization ». In : *SIAM Journal on Numerical Analysis* 24.5, p. 1152–1170 (cf. p. 22).

CACUCI, D. G. (1981). « Sensitivity theory for nonlinear systems. I. Nonlinear functional analysis approach ». In : *Journal of Mathematical Physics* 22.12, p. 2794–2802 (cf. p. 126).

CAIGNOT, A. (2009). « Prédiction par essais virtuels de l'amortissement dans les structures spatiales ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41).

CAUCHY, A. (1847). « Méthode générale pour la résolution des systemes d'équations simultanées ». In : *Comptes Rendus de l'Académie des Sciences* 25.1847, p. 536–538 (cf. p. 20–21).

ČERNÝ, V. (1985). « Thermodynamical approach to the traveling salesman problem : An efficient simulation algorithm ». In : *Journal of Optimization Theory and Applications* 45.1, p. 41–51 (cf. p. 30).

CHAMPANEY, L. (1996). « Une nouvelle approche modulaire pour l'analyse d'assemblages de structures tridimensionnelles ». Thèse de doct. Ecole Normale Supérieure de Cachan (cf. p. 40–41, 49, 51–52, 92).

— (2011). « An efficient strategy for accelerating optimization procedures for assemblies of structures with contact and friction ». In : *ICCCM 2011 - II International Conference on Computational Contact Mechanics* (cf. p. 60).

— (2012). « A domain decomposition method for studying the effects of missing fasteners on the behavior of structural assemblies with contact and friction ». In : *Computer Methods in Applied Mechanics and Engineering* 205–208. Special Issue on Advances in Computational Methods in Contact Mechanics dedicated to the memory of Professor J.A.C. Martins, p. 121–129 (cf. p. 60).

CHAMPANEY, L., C. BLANZÉ et P.-A. BOUCARD (2005). « Outils de dimensionnement et d'analyse de sensibilité pour les assemblages de structures ». In : *Mécanique & Industries* 6 (01), p. 89–99 (cf. p. 59).

CHAMPANEY, L. et P.-A. BOUCARD (2003). « Multiresolution strategy for the parametric study of assemblies including contact with friction ». In : *7th International Conference on Computational Plasticity (COMPLAS), Barcelone, Espagne* (cf. p. 41).

— (2007). « Sur la prise en compte des variations de chargement et de géométrie dans les assemblages ». In : *18e Congrés Français de Mécanique, 27–31 août, Grenoble* (cf. p. 60).

CHAMPANEY, L., P.-A. BOUCARD et S. GUINARD (2008). « Adaptive multi-analysis strategy for contact problems with friction ». In : *Computational Mechanics* 42.2, p. 305–315 (cf. p. 59).

CHAPMAN, A. et Y. SAAD (1997). « Deflated and Augmented Krylov Subspace Techniques ». In : *Numerical Linear Algebra with Applications* 4.1, p. 43–66 (cf. p. 59).

CHATTERJEE, A. (2000). « An introduction to the proper orthogonal decomposition ». In : *Current Science* 78.7, p. 808–817 (cf. p. 58).

CHATTOPADHYAY, A., T. MCCARTHY et N. PAGALDIPTI (1995). « Multilevel decomposition procedure for efficient design optimization of helicopter rotor blades ». In : *AIAA journal* 33.2, p. 223–230 (cf. p. 34).

CHAUVET, P. (1999). « Aide mémoire de la géostatistique linéaire ». In : *Cahiers de Géostatistique, école Nationale Supérieur des Mines de Paris, Centre de Géostatistique, Fontainebleau* Fascicule 2 (cf. p. 136, 138, 163).

CHEN, T. et C. YANG (2005). « Multidisciplinary design optimization of mechanisms ». In : *Advances in engineering software* 36.5, p. 301–311 (cf. p. 34, 36).

CHEN, X. (1999). « Methods of feasible directions for nonlinear programming : theory and algorithms ». Thèse de doct. Clemson University (cf. p. 24).

CHEN, X. et M. M. KOSTREVA (1999). « A generalization of the norm-relaxed method of feasible directions ». In : *Applied Mathematics and Computation* 102.2–3, p. 257–272 (cf. p. 24).

CHINESTA, F., P. LADEVÈZE et E. CUETO (2011). « A Short Review on Model Order Reduction Based on Proper Generalized Decomposition ». In : *Archives of Computational Methods in Engineering* 18.4, p. 395–404 (cf. p. 58).

CHINESTA, F., A. LEYGUE et al. (2013). « PGD-Based Computational Vademecum for Efficient Design, Optimization and Control ». In : *Archives of Computational Methods in Engineering* 20.1, p. 31–59 (cf. p. 58, 232).

CHOI, K. K., B. D. YOUN et R.-J. YANG (2001). « Moving least square method for reliability-based design optimization ». In : *Proceedings of the 4th World Congress of Structural and Multidisciplinary Optimization, Dalian* (cf. p. 79).

CHUNG, H.-S. et J. J. ALONSO (2002a). « Design of a Low-Boom Supersonic Business Jet Using Cokriging Approximation Models ». In : *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia*. AIAA 2002–5598 (cf. p. 162).

— (2002b). « Using gradients to construct cokriging approximation models for high-dimensional design optimization problems ». In : *40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada*. AIAA 2002–0317 (cf. p. 162).

CLERC, M. (2005). *L'optimisation par essaims particulaires : versions paramétriques et adaptatives*. Hermes-Lavoisier (cf. p. 29).

COGNARD, J.-Y. (1989). « Une Nouvelle approche des problémes de plasticité et de viscoplasticité la méthode à grand incrément de temps ». Thèse de doct. Université Pierre et Marie Curie, 1 vol. (187 f.) (Cf. p. 41).

COGNARD, J.-Y., P. LADEVÈZE et P. TALBOT (1999). « A large time increment approach for thermo-mechanical problems ». In : *Advances in Engineering Software* 30.9–11, p. 583–593 (cf. p. 41).

COHEN, S. et N. INTRATOR (2000). *Global Optimization of RBF Networks*. URL : `http://www.cs.tau.ac.il/~nin/papers/rbf.pdf` (cf. p. 156).

CONCEIÇÃO ANTÓNIO, C. (2002). « A multilevel genetic algorithm for optimization of geometrically nonlinear stiffened composite structures ». In : *Structural and Multidisciplinary Optimization* 24.5, p. 372–386 (cf. p. 36).

COURRIER, N., P.-A. BOUCARD et B. SOULIER (2013). « Utilisation de calculs partiellement convergés pour la construction de métamodéle ». In : *11e Coloque National en Calculs des Structures, Giens, France* (cf. p. 60, 232).

COX, D. D. et S. JOHN (1997). « SDO : A Statistical Method for Global Optimization ». In : *Multidisciplinary Design Optimization : State-of-the-Art*, p. 315–329 (cf. p. 207).

CRESSIE, N. A. C. (1993). *Statistics for spatial data, Revised Edition*. John Wiley & Sons, Ltd, Chichester (cf. p. 77, 135, 149).

CURRIN, C. et al. (1991). « Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments ». In : *Journal of the American Statistical Association* 86.416, p. 953–963 (cf. p. 135).

DANWE, R. (1993). « une Méthode simplifiée pour l'étude d'assemblage de structures ». Thèse de doct. école Normale Supérieure de Cachan, 146 p. (Cf. p. 41).

DAVIDON, W. C. (1959). *Variable metric method for minimization*. Rapp. tech. Rep. ANL-5990 Rev. Argonne National Laboratories (cf. p. 22).

DENNIS Jr, J. E. et V. TORCZON (1991). « Direct search methods on parallel machines ». In : *SIAM Journal on Optimization* 1.4, p. 448–474 (cf. p. 20).

DEODATIS, G. (1989). « Stochastic {FEM} sensitivity analysis of nonlinear dynamic problems ». In : *Probabilistic Engineering Mechanics* 4.3, p. 135–141 (cf. p. 55).

DERUMAUX, M. (2004). « Sur la modélisation / simulation de liaisons soumises à des chocs pyrotechniques ». Thèse de doct. école Normale Supérieure de Cachan, 153 p. (Cf. p. 41).

DORIGO, M. (1992). « Optimization, learning and natural algorithms ». Thèse de doct. Politecnico di Milano, Italy (cf. p. 28).

DORIGO, M. et G. DI CARO (1999). « Ant colony optimization : a new meta-heuristic ». In : *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*. T. 2. IEEE (cf. p. 28).

DORIGO, M. et L. M. GAMBARDELLA (1997). « Ant colony system : A cooperative learning approach to the traveling salesman problem ». In : *IEEE Transactions on Evolutionary Computation* 1.1, p. 53–66 (cf. p. 28–29).

DOUCHIN, B. (2000). « Stratégie de calcul robuste et adaptative pour la simulation de l'endommagement et de la rupture des structures composites stratifiées ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41).

DRÉO, J. (2004). « Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédical ». Thèse de doct. Université Paris XII - Val de Marne (cf. p. 29, 31–32).

DUCHON, J. (1977). « Splines minimizing rotation-invariant semi-norms in Sobolev spaces ». In : *Constructive Theory of Functions of Several Variables*. Sous la dir. de W. SCHEMPP et K. ZELLER. T. 571. Lecture Notes in Mathematics. Springer Berlin Heidelberg, p. 85–100 (cf. p. 156).

DUMONTET, J. et J. VIGNES (1977). « Détermination du pas optimal dans le calcul des dérivées sur ordinateur ». In : *RAIRO – Analyse numérique* 11.1, p. 13–25 (cf. p. 122).

DUREISSEIX, D. (1997). « Une approche multi-échelles pour des calculs de structures sur ordinateurs à architecture paralléle ». Thése de doctorat Sciences appliquées Cachan, école normale supérieure 1997. Thèse de doct., 1 vol. (190 p.) (Cf. p. 41).

DUYSINX, P. (1996). *Introduction à l'optimisation topologique*. Rapp. tech. OF-43. LTAS, Université de Liége (cf. p. 12–13).

DWIGHT, R. et Z.-H. HAN (2009). « Efficient Uncertainty Quantification Using Gradient-Enhanced Kriging ». In : *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics et Astronautics (cf. p. 162).

ELWAKEIL, O. A. et J. S. ARORA (1995). « Methods for finding feasible points in constrained optimization ». In : *AIAA Journal* 33.9, p. 1715–1719 (cf. p. 24).

ENGELS, H., W. BECKER et A. MORRIS (2004). « Implementation of a multi-level optimisation methodology within the e-design of a blended wing body* 1 ». In : *Aerospace Science and technology* 8.2, p. 145–153 (cf. p. 33–34, 38).

FANG, K.-T. (1980). « The uniform design : application of number-theoretic methods in experimental design ». In : *Acta Math. Appl. Sinica* 3, p. 363–372 (cf. p. 73).

FANG, K.-T. et al. (2000). « Uniform Design : Theory and Application ». In : *Technometrics* 42.3, p. 237–248 (cf. p. 73).

FARHAT, C., M. LESOINNE et K. PIERSON (2000). « A scalable dual-primal domain decomposition method ». In : *Numerical Linear Algebra with Applications* 7.7-8, p. 687–714 (cf. p. 45).

FARHAT, C. et F.-X. ROUX (1991). « A method of finite element tearing and interconnecting and its parallel solution algorithm ». In : *International Journal for Numerical Methods in Engineering* 32.6, p. 1205–1227 (cf. p. 44).

FASSHAUER, G. et J. ZHANG (2007). « On choosing "optimal" shape parameters for RBF approximation ». In : *Numerical Algorithms* 45.1-4, p. 345–368 (cf. p. 159).

FIACCO, A. V. et G. P. MCCORMICK (1968). *Nonlinear programming : Sequential unconstrained minimization techniques*. John Wiley & Sons Inc., New York, NY (cf. p. 25).

FIKE, J. A. et J. J. ALONSO (2011). « The Development of Hyper-Dual Numbers for Exact Second-Derivative Calculations ». In : *AIAA paper 2011-886, 49th AIAA Aerospace Sciences Meeting* (cf. p. 124).

FLETCHER, R. et M. J. D. POWELL (1963). « A Rapidly Convergent Descent Method for Minimization ». In : *The Computer Journal* 6.2, p. 163–168 (cf. p. 22).

FLETCHER, R. et C. M. REEVES (1964). « Function minimization by conjugate gradients ». In : *The Computer Journal* 7.2, p. 149–154 (cf. p. 21).

FLETCHER, R. (1970). « A new approach to variable metric algorithms ». In : *The computer journal* 13.3, p. 317–322 (cf. p. 22).

FOGEL, L. J. (1962). « Autonomous automata ». In : *Industrial Research* 4.2, p. 14–19 (cf. p. 28).

FOGEL, L. J. (1964). « On the organization of intellect ». Thèse de doct. University of California, Los Angeles-Engineering (cf. p. 28).

FONTAN, M. (2011). « Identification de paramétres par analyse inverse à l'aide dun algorithme méta-heuristique : applications à l'interaction sol structure, à la caractérisation de défauts et à l'optimisation de la métrologie ». Thèse de doct. Université Bordeaux 1 (cf. p. 29).

FORRESTER, A. I. J., A. J. KEANE et N. W. BRESSLOFF (2006). « Design and Analysis of "Noisy" Computer Experiments ». In : *AIAA Journal* 44.10, p. 2331–2339 (cf. p. 154).

FORRESTER, A. I. J., A. SÓBESTER et A. J. KEANE (2006). « Optimization with missing data ». In : *Proceedings of the Royal Society A : Mathematical, Physical and Engineering Science* 462.2067, p. 935–945 (cf. p. 232).

FORRESTER, A. I. J., A. SÓBESTER et A. J. KEANE (2008). *Engineering Design via Surrogate Modelling : A Practical Guide*. 1re éd. Wiley (cf. p. 64–65, 83, 150, 171, 188).

FORSGREN, A., P. GILL et M. H. WRIGHT (2002). « Interior Methods for Nonlinear Optimization ». In : *SIAM Review* 44.4, p. 525–597 (cf. p. 25).

FRANCO, J. (2008). « Planification d'expériences numériques en phase exploratoire pour la simulation des phénoménes complexes ». Thèse de doct. école Nationale Supérieure des Mines de Saint-étienne (cf. p. 66–67, 73–74).

FRANKE, R. (1982). « Scattered Data Interpolation : Tests of Some Methods. » In : *Math. Comput.* 38.157, p. 181–200 (cf. p. 159).

FRIEDMAN, J. H. (1991). « Multivariate Adaptive Regression Splines ». In : *The Annals of Statistics* 19.1, pages (cf. p. 77).

FRIMANNSLUND, L. et T. STEIHAUG (2007). « A generating set search method using curvature information ». In : *Computational Optimization and Applications* 38.1, p. 105–121 (cf. p. 20).

FRISCH, K. R. (1955). *The logarithmic potential method of convex programming*. Rapp. tech. Memorandum of May 13. The logarithmic potential method of convex programming (cf. p. 25).

GARCIA PALOMARES, U. M. et O. L. MANGASARIAN (1976). « Superlinearly convergent quasi-newton algorithms for nonlinearly constrained optimization problems ». In : *Mathematical Programming* 11.1, p. 1–13 (cf. p. 25).

GAUCHI, J. (2005). « Plans d'expériences optimaux : un exposé didactique ». In : *Modulad* 33, p. 139–162 (cf. p. 75).

GEISSER, S. (1975). « The Predictive Sample Reuse Method with Applications ». In : *Journal of the American Statistical Association* 70.350, pages (cf. p. 87).

GHANEM, R. G. et P. D. SPANOS (1991). *Stochastic Finite Elements : A Spectral Approach*. Springer-Verlag (cf. p. 56).

GIANNAKOGLOU, K., D. PAPADIMITRIOU et I. KAMPOLIS (2006). « Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels ». In : *Computer methods in applied mechanics and engineering* 195.44-47, p. 6312–6329 (cf. p. 171).

GIBBS, M. N. (1997). « Bayesian Gaussian processes for regression and classification ». Thèse de doct. University of Cambridge (cf. p. 159).

GILBERT, J. C. et J. NOCEDAL (1992). « Global convergence properties of conjugate gradient methods for optimization ». In : *SIAM Journal on Optimization* 2.1, p. 21–42 (cf. p. 21).

GILES, M. B. et N. A. PIERCE (2000). « An Introduction to the Adjoint Approach to Design ». In : *Flow, Turbulence and Combustion* 65.3-4, p. 393–415 (cf. p. 126).

GILL, P., W. MURRAY et M. WRIGHT (1981). *Practical optimization*. Academic Press, London et New York (cf. p. 123).

GILL, P. et al. (1981). *A Procedure for Computing Forward-Difference Intervals for Numerical Optimization.* Rapp. tech. SOL-81-25. Systems Optimization Laboratory, Stanford university, CA (cf. p. 122).

— (1983). « Computing forward-difference intervals for numerical optimization ». In : *SIAM Journal on Scientific and Statistical Computing* 4, p. 310 (cf. p. 122–123).

GINGOLD, R. A. et J. J. MONAGHAN (1977). « Smoothed particle hydrodynamics- Theory and application to non-spherical stars ». In : *Royal Astronomical Society, Monthly Notices* 181, p. 375–389 (cf. p. 80).

GINSBOURGER, D. (2009). « Multiples métamodéles pour l'approximation et l'optimisation de fonctions numériques multivariables ». Thèse de doct. école Nationale Supérieure des Mines de Saint-étienne (cf. p. 154, 208, 211–212).

GIUNTA, A. A., J. M. DUDLEY et al. (1994). « Noisy Aerodynamic Response And Smooth Approximations In HSCT design ». In : *In Proceedings of the Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, FL, September 7–9*, p. 1117–1128 (cf. p. 78).

GIUNTA, A. A., L. T. WATSON et J. R. KOEHLER (1998). « A comparison of approximation modeling techniques : polynomial versus interpolating models ». In : *AIAA paper*, p. 98–4758 (cf. p. 89).

GLOVER, F. (1986). « Future paths for integer programming and links to artificial intelligence ». In : *Computers & Operations Research* 13.5, p. 533–549 (cf. p. 29).

GLOVER, F. et E. TAILLARD (1993). « A user's guide to tabu search ». In : *Annals of operations research* 41.1, p. 1–28 (cf. p. 29).

GLOWINSKI, R. et P. LE TALLEC (1990). « Augmented Lagrangian interpretation of the nonoverlapping Schwarz alternating method ». In : *Third International symposium on Domain Decomposition Methods for Partial Differential Equations*. SIAM, p. 224–231 (cf. p. 45).

GOLDBERG, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc. (cf. p. 28).

GOLDFARB, D. (1970). « A family of variable-metric methods derived by variational means ». In : *Mathematics of computation* 24.109, p. 23–26 (cf. p. 22).

GOLDSTEIN, A. A. (1964). « Convex programming in Hilbert space ». In : *Bull. Amer. Math. Soc.* 70, p. 709–710 (cf. p. 24).

GOOVAERTS, P. (1997). *Geostatistics for natural resources evaluation*. Oxford University Press, USA (cf. p. 161).

GOSSELET, P. et C. REY (2006). « Non-overlapping domain decomposition methods in structural mechanics ». In : *Archives of Computational Methods in Engineering* 13.4, p. 515–572 (cf. p. 45).

GRIEWANK, A. (1989). « On automatic differentiation ». In : *Mathematical Programming : recent developments and applications*. Amsterdam, p. 83–108 (cf. p. 126).

GUIDAULT, P.-A. (2005). « Une stratégie de calcul pour les structures fissurées analyse locale-globale et approche multiéchelle pour la fissuration ». Thèse de doct. école Normale Supérieure de Cachan, 1 vol. (225 p.) (Cf. p. 41).

GUINARD, S. et al. (2002). « A 3D damage analysis of low-velocity impacts on laminated composites ». In : *Composites Science and Technology* 62.4, p. 585–589 (cf. p. 41).

GUNN, S. R. (1998). *Support Vector Machines for Classification and Regression*. Rapp. tech. University of Southampton (cf. p. 83, 85).

GUNZBURGER, M. et J. BURKARDT (2004). *Uniformity measures for point sample in hypercubes*. Rapp. tech. Florida State University (cf. p. 73).

HAFTKA, R. T. et H. M. ADELMAN (1989). « Recent developments in structural sensitivity analysis ». In : *Structural Optimization* 1.3, p. 137–151 (cf. p. 119, 125).

HALTON, J. H. (1960). « On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals ». In : *Numerische Mathematik* 2.1, p. 84–90 (cf. p. 73).

HAMMERSLEY, J. M. (1960). « Monte-Carlo methods for solving multivariate problems ». In : *Annals of the New York Academy of Sciences* 86.3, p. 844–874 (cf. p. 73).

HAMMING, R. W. (1973). *Numerical Methods for Scientists and Engineers*. 2nd. McGraw-Hill, Inc., New York, NY (cf. p. 123).

HAN, S.-P. (1977). « A globally convergent method for nonlinear programming ». In : *Journal of optimization theory and applications* 22.3, p. 297–309 (cf. p. 25).

HAN, Z.-H., S. GÖRTZ et R. ZIMMERMANN (2013). « Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function ». In : *Aerospace Science and Technology* 25.1, p. 177–189 (cf. p. 162).

HARDY, R. L. (1971). « Multiquadric equations of topography and other irregular surfaces ». In : *Journal of Geophysical Research* 76.8, p. 1905–1915 (cf. p. 6, 156, 159).

HASCOËT, L. (2004). « TAPENADE : a tool for Automatic Differentiation of programs ». In : *Proceedings of 4th European Congress on Computational Methods, ECCOMAS'2004, Jyvaskyla, Finland* (cf. p. 127).

HASCOËT, L. et M. ARAYA-POLO (2006). « Enabling User-driven checkpointing strategies in Reverse-mode Automatic Differentiation ». In : *Proceedings of the ECCOMAS CFD 2006 conference,Egmond aan Zee, The Netherlands*. also INRIA Research Report #5930 (cf. p. 127).

HAYKIN, S. (1994). *Neural networks : a comprehensive foundation*. Prentice Hall PTR Upper Saddle River, NJ, USA (cf. p. 159).

HEBB, D. (1949). *The organization of behavior*. New York : John Wiley, p. 43–54 (cf. p. 76, 81).

HESTENES, M. R. et E. STIEFEL (1952). « Methods of Conjugate Gradients for Solving Linear Systems ». In : *Journal of Research of the National Bureau of Standards* 49, p. 409–436 (cf. p. 21).

HEYBERGER, C., P.-A. BOUCARD et D. NÉRON (2012). « Multiparametric analysis within the proper generalized decomposition framework ». In : *Computational Mechanics* 49.3, p. 277–289 (cf. p. 41, 58, 60).

HEYBERGER, C., P.-A. BOUCARD et D. NÉRON (2013). « A rational strategy for the resolution of parametrized problems in the PGD framework ». In : *Computer Methods in Applied Mechanics and Engineering* 259, p. 40–49 (cf. p. 58, 60).

HISADA, T. (1988). « Sensitivity analysis of nonlinear fem ». In : *Probabilistic Methods in Civil Engineering*. ASCE, p. 160–163 (cf. p. 55).

HISADA, T. et S. NAKAGIRI (1985). « Role of stochastic finite element method in structural safety and reliability ». In : *Proc. 4th. Int. Conf. Struct. Safety and Reliability, ICOSSAR'85*, p. 385–395 (cf. p. 54).

HOEF, J. M. et N. A. C. CRESSIE (1993). « Multivariable spatial prediction ». In : *Mathematical Geology* 25.2, p. 219–240 (cf. p. 161).

HOLLAND, J. H. (1962). « Outline for a Logical Theory of Adaptive Systems ». In : *Journal of the ACM (JACM)* 9.3, p. 297–314 (cf. p. 28).

HOLLAND, J. H. (1975). *Adaptation in natural and artificial systems an introductory analysis with applications to biology, control, and artificial intelligence*. The University of Michigan Press (cf. p. 28).

HOOKE, R. et T. A. JEEVES (1961). « " Direct Search" Solution of Numerical and Statistical Problems ». In : *Journal of the ACM (JACM)* 8.2, p. 212–229 (cf. p. 20).

HUANG, D. et al. (2006). « Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models ». In : *Journal of Global Optimization* 34.3, p. 441–466 (cf. p. 211–213, 220).

IOTT, J., R. T. HAFTKA et H. M. ADELMAN (1985). *Selecting step sizes in sensitivity analysis by finite differences*. Rapp. tech. Technical Memorandum 86382. NASA (cf. p. 123).

ISAAKS, E. et R. SRIVASTAVA (1989). *An introduction to applied geostatistics*. Oxford University Press, New York (cf. p. 161).

JAMESON, A. (1988). « Aerodynamic design via control theory ». In : *Journal of Scientific Computing* 3.3, p. 233–260 (cf. p. 126).

— (2001). « A perspective on computational algorithms for aerodynamic analysis and design ». In : *Progress in Aerospace Sciences* 37.2, p. 197–243 (cf. p. 126).

JAMESON, A. et S. KIM (2003). « Reduction of the Adjoint Gradient Formula for Aerodynamic Shape Optimization Problems ». In : *AIAA Journal* 41.11, p. 2114–2129 (cf. p. 126).

JAYADEVA, R. KHEMCHANDANI et S. CHANDRA (2006). « Regularized Least Squares Twin SVR for the Simultaneous Learning of a Function and its Derivative ». In : *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, p. 1192–1197 (cf. p. 173).

JIN, R., W. CHEN et T. SIMPSON (2000). « Comparative Studies Of Metamodeling Techniques Under Multiple Modeling Criteria ». In : *Structural and Multidisciplinary Optimization* 23, p. 1–13 (cf. p. 89).

JIN, R., X. DU et W. CHEN (2003). « The use of metamodeling techniques for optimization under uncertainty ». In : *Structural and Multidisciplinary Optimization* 25.2, p. 99–116 (cf. p. 88).

JIN, R., W. CHEN et A. SUDJIANTO (2002). « On sequential sampling for global metamodeling in engineering design ». In : *Proceedings of DETC, Montréal, Canada* (cf. p. 211).

JOHNSON, M., L. MOORE et D. YLVISAKER (1990). « Minimax and maximin distance designs ». In : *Journal of statistical planning and inference* 26.2, p. 131–148 (cf. p. 70, 74).

JONES, D. R. (2001). « A Taxonomy of Global Optimization Methods Based on Response Surfaces ». In : *Journal of Global Optimization* 21.4, p. 345–383 (cf. p. 204, 208, 211).

JONES, D. R., C. D. PERTTUNEN et B. E. STUCKMAN (1993). « Lipschitzian optimization without the Lipschitz constant ». In : *Journal of Optimization Theory and Applications* 79.1, p. 157–181 (cf. p. 26, 207).

JONES, D. R., M. SCHONLAU et W. J. WELCH (1998). « Efficient global optimization of expensive black-box functions ». In : *Journal of Global optimization* 13.4, p. 455–492 (cf. p. 66, 209, 211–213, 228).

JONES, R. H. et A. V. VECCHIA (1993). « Fitting continuous ARMA models to unequally spaced spatial data ». In : *Journal of the American Statistical Association* 88.423, p. 947–954 (cf. p. 149).

JOURDAN, A. (2000). « Analyse statistique et échantillonnage d'expériences simulées ». Thèse de doct. Université de Pau et des Pays de l'Adour (cf. p. 67, 73).

KALAGNANAM, J. R. et U. M. DIWEKAR (1997). « An Efficient Sampling Technique for Off-line Quality Control ». In : *Technometrics* 39.3, p. 308–319 (cf. p. 73).

KAMPOLIS, I. C., E. I. KARANGELOS et K. C. GIANNAKOGLOU (2004). « Gradient-assisted radial basis function networks : theory and applications ». In : *Applied Mathematical Modelling* 28.2, p. 197–209 (cf. p. 171).

KARMARKAR, N. (1984). « A new polynomial-time algorithm for linear programming ». In : *Combinatorica* 4.4, p. 373–395 (cf. p. 25, 85).

KARUSH, W. (1939). « Minima of functions of several variables with inequalities as side constraints ». Mém.de mast. Dept. of Mathematics, University of Chicago (cf. p. 17).

KEANE, A. J. et P. B. NAIR (2005). *Computational Approaches for Aerospace Design : The Pursuit of Excellence*. John Wiley & Sons, Ltd., Hoboken, England (cf. p. 125–126).

KEANE, A. J. et N. PETRUZZELLI (2000). « Aircraft wing design using GA-based multi-level strategies ». In : *AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, USA, 06 - 08 Sep 2000*. AIAA 2000-4937 (cf. p. 37).

KENNEDY, J. et R. C. EBERHART (1995). « Particle swarm optimization ». In : *Proceedings of IEEE International Conference on Neural Networks*. T. 4, p. 1942–1948 (cf. p. 29).

KENNEDY, M. et A. O'HAGAN (2000). « Predicting the output from a complex computer code when fast approximations are available ». In : *Biometrika* 87.1, p. 1 (cf. p. 135).

KERFRIDEN, P. (2008). « Stratégie de décomposition de domaine à trois échelles pour la simulation du délaminage dans les stratifiés ». Thèse de doct. école Normale Supérieure de Cachan, 1 vol. (163 p.) (Cf. p. 41).

KIM, B.-S., Y.-B. LEE et D.-H. CHOI (2009). « Comparison study on the accuracy of metamodeling technique for non-convex functions ». In : *Journal of Mechanical Science and Technology* 23.4, p. 1175–1181 (cf. p. 89).

KIM, C., S. WANG et K. K. CHOI (2005). « Efficient Response Surface Modeling by Using Moving Least-Squares Method and Sensitivity ». In : *AIAA Journal* 43.11, p. 2404–2411 (cf. p. 173).

KIRKPATRICK, S., C. D. GELATT et M. P. VECCHI (1983). « Optimization by Simulated Annealing ». In : *Science* 220.4598, p. 671–680 (cf. p. 30).

KITANIDIS, P. K. (1983). « Statistical estimation of polynomial generalized covariance functions and hydrologic applications ». In : *Water Resources Research* 19.4, p. 909–921 (cf. p. 149).

KLEIJNEN, J. P. et R. G. SARGENT (2000). « A methodology for fitting and validating metamodels in simulation ». In : *European Journal of Operational Research* 120.1, p. 14–29 (cf. p. 62).

KOEHLER, J. R. et A. B. OWEN (1996). « Computer experiments ». In : *Handbook of statistics* 13, p. 261–308 (cf. p. 67, 162).

KOLDA, T. G., R. M. LEWIS et V. TORCZON (2003). « Optimization by Direct Search : New Perspectives on Some Classical and Modern Methods ». In : *SIAM Review* 45.3, p. 385–482 (cf. p. 20).

KRAVANJA, S., S. ŠILIH et Z. KRAVANJA (2005). « The multilevel {MINLP} optimization approach to structural synthesis : the simultaneous topology, material, standard and rounded dimension optimization ». In : *Advances in Engineering Software* 36.9. Design Optimization, p. 568–583 (cf. p. 35).

KRAVANJA, S., A. SORŠAK et Z. KRAVANJA (2003). « Efficient Multilevel MINLP Strategies for Solving Large Combinatorial Problems in Engineering ». In : *Optimization and Engineering* 4.1-2, p. 97–151 (cf. p. 35).

KRIGE, D. G. (1951a). « A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand ». In : *Journal of the Chemical, Metallurgical and Mining Society of South Africa* 52.6, p. 119–139 (cf. p. 76, 134).

— (1951b). « A statistical approach to some mine valuation and allied problems on the Witwatersrand ». Mém.de mast. University of the Witwatersrand (cf. p. 5, 76).

KUHN, H. W. et A. W. TUCKER (1951). « Nonlinear programming ». In : *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950*. Berkeley et Los Angeles : University of California Press, p. 481–492 (cf. p. 17).

KUSHNER, H. J. (1964). « A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise ». In : *Journal of Basic Engineering* 86.1, p. 97–106 (cf. p. 208).

LADEVÈZE, P. (1985). « Sur une famille d'algorithmes en mécanique des structures ». In : *Comptes Rendus de l'Académie des Sciences* 300.2, p. 41–44 (cf. p. 4, 39, 45, 52, 57–58).

— (1987). « Sur une mécanique paralléle : concepts et outils de calculs ». In : *Séminaire du Laboratoire de Mécanique et Technologie de Cachan* (cf. p. 41).

— (1990). *Une nouvelle méthode pour l'analyse des structures massives*. Rapp. tech. STS/T3 n °45 465. AÉROSPATIALE (cf. p. 40).

— (1999). *Nonlinear computational structural mechanics : new approaches and non-incremental methods of calculation*. Mechanical Engineering Series. Springer Verlag (cf. p. 4, 40, 45, 49).

LADEVÈZE, P. et D. DUREISSEIX (1999). « Une nouvelle stratégie de calcul micro/macro en mécanique des structures ». In : *Comptes Rendus de l'Académie des Sciences - Series {IIB} - Mechanics-Physics-Astronomy* 327.12, p. 1237–1244 (cf. p. 41).

LADEVÈZE, P., O. LOISEAU et D. DUREISSEIX (2001). « A micro–macro and parallel computational strategy for highly heterogeneous structures ». In : *International Journal for Numerical Methods in Engineering* 52.1-2, p. 121–138 (cf. p. 52).

LANCASTER, P. et K. SALKAUSKAS (1981). « Surfaces generated by moving least squares methods ». In : *Mathematics of computation* 37.155, p. 141–158 (cf. p. 79).

LANCASTER, P. et K. ŠALKAUSKAS (1986). *Curve and surface fitting : An introduction*. London San Diego New York [etc.] : Academic Press (cf. p. 79).

LAND, A. H. et A. G. DOIG (1960). « An Automatic Method of Solving Discrete Programming Problems ». In : *Econometrica* 28.3, p. 497–520 (cf. p. 26).

LAURENCEAU, J. et M. MEAUX (2008). « Comparison of Gradient and Response Surface Based Optimization Frameworks Using Adjoint Method ». In : *AIAA Paper* 2008-1889 (cf. p. 162, 170).

LAURENCEAU, J. et P. SAGAUT (2008). « Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging ». In : *AIAA journal* 46.2, p. 498 (cf. p. 162).

LÁZARO, M. et al. (2005). « Support Vector Regression for the simultaneous learning of a multivariate function and its derivatives ». In : *Neurocomputing* 69.1–3, p. 42–61 (cf. p. 173).

LE RICHE, R. et J. GAUDIN (1998). « Design of dimensionally stable composites by evolutionary optimization ». In : *Composite Structures* 41.2, p. 97–111 (cf. p. 36).

LEARY, S. J., A. BHASKAR et A. J. KEANE (2004a). « A derivative based surrogate model for approximating and optimizing the output of an expensive computer simulation ». In : *Journal of Global Optimization* 30.1, p. 39–58 (cf. p. 162, 173).

— (2004b). « Global approximation and optimization using adjoint computational fluid dynamics codes ». In : *AIAA journal* 42.3, p. 631–641 (cf. p. 162, 171, 205).

LEMOUSSU, H. (2000). « Sur une approche pour l'analyse et le calcul de structures 3D soumises à un choc pyrotechnique ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41).

LEVITIN, E. et B. T. POLYAK (1966). « Constrained minimization problems ». In : *USSR Computational Mathematics and Mathematical Physics* 6, p. 1–50 (cf. p. 24).

LEVY, A. et A. MONTALVO (1985). « The Tunneling Algorithm for the Global Minimization of Functions ». In : *SIAM Journal on Scientific and Statistical Computing* 6.1, p. 15–29 (cf. p. 27).

LEWIS, G. A., D. MATHIEU et R. PHAN-TAN-LUU (1999). *Pharmaceutical Experimental Design*. Marcel Dekker Inc., New York (cf. p. 68).

LEWIS, R. M. (1998). « Using sensitivity information in the construction of kriging models for design optimization ». In : *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis & Optimization Symposium, Saint Louis, Missouri* (cf. p. 162).

LEWIS, R. M., V. TORCZON et M. W. TROSSET (2000). « Direct search methods : then and now ». In : *Journal of Computational and Applied Mathematics* 124.1–2, p. 191–207 (cf. p. 20).

LINDLEY, D. V. (1956). « On a Measure of the Information Provided by an Experiment ». In : *The Annals of Mathematical Statistics* 27.4, pages (cf. p. 75).

LIONS, J.-L. (1971). *Optimal Control of Systems Governed by Partial Dierential Equations.* Springer Verlag, New York (cf. p. 126).

LIU, B. (1992). « Simulation numérique de l'emboutissage. Méthode a grand incrément de temps ». Thèse de doct. Université Pierre et Marie Curie, 171 p. (Cf. p. 41).

LIU, B., R. T. HAFTKA et L. T. WATSON (2004). « Global-local structural optimization using response surfaces of local optimization margins ». In : *Structural and Multidisciplinary Optimization* 27.5, p. 352–359 (cf. p. 35).

LIU, J., Z.-H. HAN et W. P. SONG (2012). « Comparison of Infill Sampling Criteria in Kriging-Based Aerodynamic Optimization ». In : *28th Congress of the International Council of the Aeronautical Sciences, 23 - 28 September, Brisbane, Australia* (cf. p. 211).

LIU, W. (2003). « Development of Gradient-Enhanced Kriging Approximations for Multidisciplinary Design Optimization ». Thèse de doct. University of Notre Dame, Indiana (cf. p. 170).

LOCKWOOD, B. A. et M. ANITESCU (2012). « Gradient-enhanced universal kriging for uncertainty propagation ». In : *Nuclear Science and Engineering* 170.2, p. 168–195 (cf. p. 168).

LOCKWOOD, B. A. et D. J. MAVRIPLIS (2010). « Parameter Sensitivity Analysis for Hypersonic Viscous Flow using a Discrete Adjoint Approach ». In : *AIAA Paper* 447 (cf. p. 162).

LOÉVE, M. (1977). *Probability theory*. Springer-Verlag, New York (cf. p. 56).

LOISEAU, O. (2001). « Une stratégie de calcul multiéchelle pour les structures hétérogénes ». Thèse de doct. Ecole Normale Supérieure de Cachan (cf. p. 41).

LOPHAVEN, S. N., H. B. NIELSEN et J. SØNDERGAARD (2002). *DACE : a Matlab Kriging toolbox, version 2.0*. Rapp. tech. IMM-REP-2002-12. Informatics et Mathematical Modelling, DTU (cf. p. 5, 150, 154).

LORONG, P. (1994). « Sur une approche paralléle pour le calcul des structures. Comportement sur calculateurs paralléles ». Thèse de doct. école Normale Supérieure de Cachan, 152 P. (Cf. p. 41).

LUMLEY, J. (1967). « The structure of inhomogeneous turbulent flows ». In : *Atmospheric Turbulence and Radio Wave Propagation*. Sous la dir. d'I. A.M. et V. TATARSKI. Nauka, Moscow, p. 221–227 (cf. p. 57).

LYNESS, J. (1977). « Has numerical differentiation a future ? » In : *Proceedings 7th Manitoba Conference on Numerical Mathematics*. Utilitas Mathematica Publishing, p. 107–129 (cf. p. 122).

LYNESS, J. et C. MOLER (1967). « Numerical Differentiation of Analytic Functions ». In : *SIAM Journal on Numerical Analysis* 4.2, p. 202–210 (cf. p. 124).

MADAY, Y., A. T. PATERA et G. TURINICI (2002). « A Priori Convergence Theory for Reduced-Basis Approximations of Single-Parameter Elliptic Partial Differential Equations ». In : *Journal of Scientific Computing* 17.1–4, p. 437–446 (cf. p. 58).

MANDEL, J. (1993). « Balancing domain decomposition ». In : *Communications in numerical methods in engineering* 9.3, p. 233–241 (cf. p. 44).

MARCELET, M. (2008). « Etude et mise en oeuvre d'une méthode d'optimisation de forme couplant simulation numérique en aérodynamique et en calcul de structure ». Thèse de doct. Ecole Nationale d'Arts et Métiers (cf. p. 89).

MARCH, A., K. WILLCOX et Q. WANG (2010). « Gradient-based multifidelity optimisation for aircraft design using Bayesian model calibration ». In : *Aeronautical Journal* 115.1174, p. 729–738 (cf. p. 162).

MARDIA, K. et R. MARSHALL (1984). « Maximum likelihood estimation of models for residual covariance in spatial regression ». In : *Biometrika* 71.1, p. 135 (cf. p. 149).

MARTINS, J. R. R. A. (2002). « Coupled-Adjoint Method for High-Fidelity Aero-Structural Optimization ». Thèse de doct. Stanford University (cf. p. 125).

MARTINS, J. R. R. A., J. J. ALONSO et J. J. REUTHER (2004). « High-Fidelity Aerostructural Design Optimization of a Supersonic Business Jet ». In : *Journal of Aircraft* 41.3, p. 523–530 (cf. p. 125).

MARTINS, J. R. R. A., P. STURDZA et J. J. ALONSO (2003). « The complex-step derivative approximation ». In : *ACM Transactions on Mathematical Software (TOMS)* 29.3, p. 245–262 (cf. p. 125).

MATÉRN, B. (1960). *Spatial Variation (Lecture NotesStatist. 36)*. Springer, Berlin (cf. p. 144).

MATHERON, G. (1962). « Traité de géostatistique appliquée, Tome I ». In : *Mémoires du Bureau de Recherches Géologiques et Miniéres* 14 (cf. p. 134).

— (1963a). « Traité de Géostatistique Appliquée, Tome II : Le Krigeage ». In : *Mémoires du Bureau de Recherches Géologiques et Miniéres* 24 (cf. p. 134).

— (1965). *Les variables régionalisées et leur estimation*. Masson (cf. p. 135).

MATHERON, G. (1963b). « Principles of geostatistics ». In : *Economic Geology* 58.8, p. 1246–1266 (cf. p. 5, 77).

— (1970). « La théorie des variables régionalisées et ses applications ». In : *Les Cahiers du Centre de Morphologie Mathématique de Fontainebleau, école Nationale des Mines de Paris* Fascicule 5 (cf. p. 136, 160).

MCCULLOCH, W. et W. PITTS (1943). « A logical calculus of the ideas immanent in nervous activity ». In : *Bulletin of Mathematical Biology* 5.4, p. 115–133 (cf. p. 76, 81).

MCKAY, M., W. CONOVER et R. BECKMAN (1979). « A comparison of three methods for selecting values of input variables in the analysis of output from a computer code ». In : *Technometrics* 21.2, p. 239–245 (cf. p. 69).

MCKINNON, K. I. (1998). « Convergence of the Nelder–Mead Simplex Method to a Nonstationary Point ». In : *SIAM Journal on Optimization* 9.1, p. 148–158 (cf. p. 20).

METROPOLIS, N. et al. (1953). « Equation of State Calculations by Fast Computing Machines ». In : *The Journal of Chemical Physics* 21.6, p. 1087–1092 (cf. p. 30).

MICHEL-PONNELLE, S. (2001). « Modélisation et simulation de structures élastoméres en dommageables en grandes transformations ». Thèse de doct. école Normale Supérieure de Cachan, 1 vol. (162 p.) (Cf. p. 41).

MOODY, J. et C. J. DARKEN (1989). « Fast learning in networks of locally-tuned processing units ». In : *Neural Comput.* 1.2, p. 281–294 (cf. p. 156).

MORÉ, J. J. et S. M. WILD (2011). « Estimating Computational Noise ». In : *SIAM Journal on Scientific Computing* 33.3, p. 1292–1314 (cf. p. 123, 128, 232).

MORÉ, J. J. et S. M. WILD (2012a). *Do You Trust Derivatives or Differences ?* Preprint ANL/MCS-P2067-0312. Argonne Mathematics et Computer Science Division (cf. p. 123, 232).

— (2012b). « Estimating Derivatives of Noisy Simulations ». In : *ACM Trans. Math. Softw.* 38.3, 19 :1–19 :21 (cf. p. 123, 232).

MORRIS, M. D. et T. J. MITCHELL (1995). « Exploratory designs for computational experiments ». In : *Journal of Statistical Planning and Inference* 43.3, p. 381–402 (cf. p. 70).

MORRIS, M. D., T. J. MITCHELL et D. YLVISAKER (1993). « Bayesian design and analysis of computer experiments : use of derivatives in surface prediction ». In : *Technometrics* 35.3, p. 243–255 (cf. p. 161–163, 167–168).

MYERS, R. H. et D. C. MONTGOMERY (1995). *Response Surface Methodology. Process and Product Optimization Using Designed Experiments*. John Wiley & Sons Inc., New York, NY (cf. p. 66, 68, 77).

NEAL, R. M. (1998). « Regression and classification using Gaussian process priors ». In : *J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, Bayesian Statistics 6*. Oxford University Press (cf. p. 135).

NEALEN, A. et T. DARMSTADT (2004). *An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation*. URL : `http://nealen.de/projects/mls/asapmls.pdf` (cf. p. 79).

NELDER, J. A. et R. MEAD (1965). « A Simplex Method for Function Minimization ». In : *The Computer Journal* 7.4, p. 308–313 (cf. p. 20).

NÉRON, D. (2004). « Sur une stratégie de calcul pour les problémes multiphysiques ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41).

NGUYEN, H.-M., O. ALLIX et P. FEISSEL (2008). « A robust identification strategy for rate-dependent models in dynamics ». In : *Inverse Problems* 24.6, p. 065006 (cf. p. 41).

NIEDERREITER, H. (1987). « Point sets and sequences with small discrepancy ». In : *Monatshefte für Mathematik* 104.4, p. 273–337 (cf. p. 73).

NOCEDAL, J. et S. J. WRIGHT (1999). *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, p. 664 (cf. p. 14, 17, 23).

NOTIN, A. et al. (2011). « SLDLT : une technique de mise à jour de la décomposition LDLT pour l'accélération des simulations de Monte Carlo dans le cas de champs stochastiques non uniformes ». In : *10e Colloque National En Calcul Des Structures, Giens, France* (cf. p. 59).

NOUY, A. (2003). « Une stratégie de calcul multiéchelle avec homogénéisation en temps et en espace pour le calcul de structures fortement hétérogénes ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41).

ODIÉVRE, D. (2009). « Sur une stratégie de calcul en dynamique transitoire en présence de variabilité paramétrique ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41, 60).

ONG, Y.-S., K. LUM et P. B. NAIR (2008). « Hybrid evolutionary algorithm with Hermite radial basis function interpolants for computationally expensive adjoint solvers ». In : *Computational Optimization and Applications* 39.1, p. 97–119 (cf. p. 171).

ORR, M. J. L. (1996). *Introduction to radial basis function networks*. Rapp. tech. University of Edinburg (cf. p. 156).

— (1999). *Recent Advances in Radial Basis Function Networks*. Rapp. tech. Institute for Adaptive et Neural Computation (cf. p. 156, 159).

OWEN, A. B. (1992). « Orthogonal arrays for computer experiments, integration and visualization ». In : *Statistica Sinica* 2.2, p. 439–452 (cf. p. 72).

— (1994). « Controlling Correlations in Latin Hypercube Samples ». In : *Journal of the American Statistical Association* 89.428, p. 1517–1522 (cf. p. 70).

PARK, J.-S. (1994). « Optimal Latin-hypercube designs for computer experiments ». In : *Journal of Statistical Planning and Inference* 39.1, p. 95–111 (cf. p. 70).

PARSOPOULOS, K. E. et M. N. VRAHATIS (2007). « Parameter selection and adaptation in Unified Particle Swarm Optimization ». In : *Mathematical and Computer Modelling* 46.1–2, p. 198–213 (cf. p. 29).

PASSIEUX, J.-C. (2008). « Approximation radiale et méthode LATIN multiéchelle en temps et espace ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41, 52).

PELLE, J.-P. et D. RYCKELYNCK (2000). « An efficient adaptive strategy to master the global quality of viscoplastic analysis ». In : *Computers & Structures* 78.1–3, p. 169–183 (cf. p. 41).

PETER, J. et al. (2007). « Comparison of surrogate models for the actual global optimization of a 2D turbomachinery flow ». In : *Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization*. World Scientific, Engineering Academy et Society (WSEAS), p. 46–51 (cf. p. 89, 205).

PIRONNEAU, O. (1974). « On optimum design in fluid mechanics ». In : *Journal of Fluid Mechanics* 64, p. 97–110 (cf. p. 126).

POWELL, M. J. (1964). « An efficient method for finding the minimum of a function of several variables without calculating derivatives ». In : *The Computer Journal* 7.2, p. 155–162 (cf. p. 20).

— (1981). *Approximation Theory and Methods*. Cambridge University Press (cf. p. 156).

— (1998). « Direct search algorithms for optimization calculations ». In : *Acta Numerica* 7, p. 287–336 (cf. p. 20).

POWELL, M. et Y. YUAN (1990). « A trust region algorithm for equality constrained optimization ». In : *Mathematical Programming* 49.1-3, p. 189–211 (cf. p. 22).

PRAVEEN, C. et R. DUVIGNEAU (2007). *Radial Basis Function and Kriging Metamodels for Aerodynamic Optimization*. Rapp. tech. 6151. INRIA (cf. p. 159).

PRZEMIENIECKI, J. S. (1963). « Matrix Structural Analysis of Substructures ». In : *AIAA Journal* 1.1, p. 138–147 (cf. p. 44).

QUEIPO, N. V. et al. (2005). « Surrogate-based analysis and optimization ». In : *Progress in Aerospace Sciences* 41.1, p. 1–28 (cf. p. 65).

RAFAJŁOWICZ, E. et R. SCHWABE (2006). « Halton and Hammersley sequences in multivariate nonparametric regression ». In : *Statistics & Probability Letters* 76.8, p. 803–812 (cf. p. 73).

RANJAN, P., R. HAYNES et R. KARSTEN (2011). « A Computationally Stable Approach to Gaussian Process Interpolation of Deterministic Computer Simulation Data ». In : *Technometrics* 53.4, p. 366–378 (cf. p. 154).

RASMUSSEN, C. E. et C. K. I. WILLIAMS (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, MA, USA : The MIT Press, p. 248 (cf. p. 135–136, 144).

RASMUSSEN, J. (1998). « Nonlinear programming by cumulative approximation refinement ». In : *Structural optimization* 15.1, p. 1–7 (cf. p. 80).

RASTRIGIN, L. (1963). « The convergence of the random search method in the extremal control of a many-parameter system ». In : *Automation and Remote Control* 24.10, p. 1337–1342 (cf. p. 26).

RECHENBERG, I. (1973). *Evolutionsstrategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog (cf. p. 28).

RELUN, N. (2011). « Approche multiparamétrique pour la conception robuste en fatigue ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41).

RELUN, N., D. NÉRON et P.-A. BOUCARD (2012). « A model reduction technique based on the PGD for elastic-viscoplastic computational analysis ». In : *Computational Mechanics*, p. 1–10 (cf. p. 60).

RINNOOY KAN, A. et G. TIMMER (1987). « Stochastic global optimization methods part I : Clustering methods ». In : *Mathematical Programming* 39.1, p. 27–56 (cf. p. 27).

RIOS, L. M. et N. V. SAHINIDIS (2013). « Derivative-free optimization : a review of algorithms and comparison of software implementations ». In : *Journal of Global Optimization* 56.3, p. 1247–1293 (cf. p. 25).

RIPPA, S. (1999). « An algorithm for selecting a good value for the parameter c in radial basis function interpolation ». In : *Advances in Computational Mathematics* 11.2, p. 193–210 (cf. p. 151, 159).

RISLER, F. et C. REY (2000). « Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems ». In : *Numerical Algorithms* 23.1, p. 1–30 (cf. p. 59).

ROBINSON, G. M. et A. J. KEANE (1999). « A case for multi-level optimisation in aeronautical design ». In : *Aeronautical Journal* 103.1028, p. 481–485 (cf. p. 37).

ROGERS, J. et W. LAMARSH (1995). « Reducing neural network training time with parallel processing ». In : *NASA Technical Memorandum 110154* (cf. p. 82).

ROSENBLATT, F. (1958). « The perceptron : A probabilistic model for information storage and organization in the brain ». In : *Psychological review* 65.6, p. 386–408 (cf. p. 76, 81).

ROSENBROCK, H. H. (1960). « An Automatic Method for Finding the Greatest or Least Value of a Function ». In : *The Computer Journal* 3.3, p. 175–184 (cf. p. 20).

ROULET, V. (2011). « Stratégie multiparamétrique pour la simulation d'assemblages de structures stratifiées ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41).

ROULET, V., P.-A. BOUCARD et L. CHAMPANEY (2013). « An efficient computational strategy for composite laminates assemblies including variability ». In : *International Journal of Solids and Structures* 50.18, p. 2749–2757 (cf. p. 60).

ROYER, C. (1990). « Une Approche des problémes de dynamique non-lineaires par la méthode à grand incrément de temps ». Thèse de doct. Université Pierre et Marie Curie, pages (cf. p. 41).

RUMPFKEIL, M. P., W. YAMAZAKI et D. J. MAVRIPLIS (2011). « A Dynamic Sampling Method for Kriging and Cokriging Surrogate Models ». In : *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics et Astronautics (cf. p. 162).

SAAD, Y. (2000). *Iterative methods for sparse linear systems*. 3e éd. PWS Publishing Compagny (cf. p. 58–59).

SAAVEDRA REDLICH, K. (2012). « Stratégie multiéchelle pour l'analyse du couplage flambage-délaminage de composites stratifiés ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41).

SACKS, J., S. B. SCHILLER et W. J. WELCH (1989). « Designs for Computer Experiments ». In : *Technometrics* 31.1, p. 41–47 (cf. p. 5, 75, 77, 135).

SACKS, J., W. J. WELCH et al. (1989). « Design and Analysis of Computer Experiments ». In : *Statistical Science* 4.4, pages (cf. p. 66–67, 75, 135, 144).

SANTNER, T. J., B. J. WILLIAMS et W. I. NOTZ (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag, p. 283 (cf. p. 135).

SASENA, M. (2002). « Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations ». Thèse de doct. University of Michigan (cf. p. 65, 211).

SCHABACK, R. (1997). *Reconstruction of Multivariate Functions from Scattered Data*. Book for teaching. URL : `http://num.math.uni-goettingen.de/schaback/teaching/texte/rbfbook.ps` (cf. p. 156, 160).

— (2007). *A Practical Guide to Radial Basis Functions*. Book for teaching. URL : `http://num.math.uni-goettingen.de/schaback/teaching/sc.pdf` (cf. p. 157).

SCHONLAU, M. (1997). « Computer Experiments and Global Optimization ». Thèse de doct. University of Waterloo (cf. p. 65, 208–209, 212).

SCHWARZ, H. (1869). « Ueber einige Abbildungsaufgaben. » In : *Journal für die reine und angewandte Mathematik* 70, p. 105–120 (cf. p. 44).

SCHWEFEL, H.-P. (1995). *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology Series. Wiley, New York (cf. p. 28).

SEN GUPTA, J. (2005). « Mésodynamique et rupture des composites 3D C/C sous choc : une stratégie numérique dédiée ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41).

SERIES, L., F. FEYEL et F.-X. ROUX (2003). « Une méthode de décomposition de domaine avec deux multiplicateurs de Lagrange ». In : *Actes du 16e Congrés Franç de Mécanique* (cf. p. 45).

SHANNO, D. F. (1970). « Conditioning of Quasi-Newton Methods for Function Minimization ». In : *Mathematics of Computation* 24.111, pages (cf. p. 22).

SHANNON, C. E. (1948). « A Mathematical Theory of Communication ». In : *The Bell System Technical Journal* 27, p. 379–423, 623–656 (cf. p. 75).

SHEWCHUK, J. R. (1994). *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Rapp. tech. Pittsburgh, PA, USA (cf. p. 21).

SHEWRY, M. C. et H. P. WYNN (1987). « Maximum entropy sampling ». In : *Journal of Applied Statistics* 14.2, p. 165–170 (cf. p. 75).

SHINOZUKA, M. (1987). « Basic issues in stochastic finite element analysis ». In : *Reliability and Risk Analysis in Civil Engineering* 1, p. 506–519 (cf. p. 54, 56).

SIMPSON, T. W., D. K. J. LIN et W. CHEN (2001). « Sampling Strategies for Computer Experiments : Design and Analysis ». In : *International Journal of Reliability and Applications* 2.3, p. 209–240 (cf. p. 67).

SIMPSON, T. et al. (1998). « Comparison Of Response Surface And Kriging Models For Multidisciplinary Design Optimization ». In : *AIAA paper 98-4758. 7 th AIAA/USAF/-NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, p. 98–4755 (cf. p. 89).

SMOLA, A. et B. SCHÖLKOPF (2004). « A tutorial on support vector regression ». In : *Statistics and Computing* 14.3, p. 199–222 (cf. p. 83–84).

SÓBESTER, A., S. J. LEARY et A. J. KEANE (2005). « On the Design of Optimization Strategies Based on Global Response Surface Approximation Models ». In : *Journal of Global Optimization* 33.1, p. 31–59 (cf. p. 66, 159, 209).

SOBOL, I. M. (1967). « On the distribution of points in a cube and the approximate evaluation of integrals ». In : *USSR Computational Mathematics and Mathematical Physics* 7.4, p. 86–112 (cf. p. 73).

SOCHA, K. et M. DORIGO (2008). « Ant colony optimization for continuous domains ». In : *European Journal of Operational Research* 185.3, p. 1155–1173 (cf. p. 29).

SOLIS, F. J. et R. J.-B. WETS (1981). « Minimization by Random Search Techniques ». In : *Mathematics of Operations Research* 6.1, p. 19–30 (cf. p. 26).

SORENSEN, D. (1982). « Trust-region methods for unconstrained minimization ». In : *Nonlinear Optimization 1981*. Sous la dir. de M. POWELL. Academic Press, London, p. 29–38 (cf. p. 22).

SOULIER, B. (2007). « Construction d'approximation globale par approche bayésienne pour l'optimisation de problémes fortement non linéaires ». In : *CFM 2007 – 18éme Congrés Français de Mécanique* (cf. p. 41).

SOULIER, B. et P.-A. BOUCARD (2009). « A multiparametric strategy for the large-scale multilevel optimization of structural assemblies ». In : *8th World Congress on Structural and Multidisciplinary Optimization, Lisbon, Portugal* (cf. p. 61–62, 110–112).

— (2012). « Coupling Partially Converged Computations and Multiparametric Strategy for Structural Optimization of Assemblies ». In : *8th European Solid Mechanics Conference* (cf. p. 60).

— (2013). « A multiparametric strategy for the two step optimization of structural assemblies ». In : *Structural and Multidisciplinary Optimization* 47.4, p. 539–553 (cf. p. 60, 80–81).

SOULIER, B., L. RICHARD et al. (2003). « Crashworthiness Optimization Using a Surrogate Approach by Stochastic Response Surface ». In : *Recent Advances in Integrated Design and Manufacturing in Mechanical Engineering*. Sous la dir. de G. GOGU et al. Springer Netherlands, p. 159–168 (cf. p. 80).

SQUIRE, W. et G. TRAPP (1998). « Using Complex Variables to Estimate Derivatives of Real Functions ». In : *SIAM Review* 40.1, p. 110–112 (cf. p. 124).

STEIN, M. L. (1990). « A Comparison of Generalized Cross Validation and Modified Maximum Likelihood for Estimating the Parameters of a Stochastic Process ». In : *The Annals of Statistics* 18.3, pages (cf. p. 151).

— (1999). *Interpolation of Spatial Data : some theory for kriging*. Springer Verlag (cf. p. 144).

STEIN, M. L., Z. CHI et L. WELTY (2004). « Approximating likelihoods for large spatial data sets ». In : *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* 66.2, p. 275–296 (cf. p. 151).

STONE, M. (1974). « Cross-validatory choice and assessment of statistical predictions ». In : *Journal of the Royal Statistical Society. Series B (Methodological)*, p. 111–147 (cf. p. 87).

TAGUCHI, G. (1986). *Introduction to quality engineering : designing quality into products and processes*. Asian Productivity Organization Tokyo (cf. p. 66, 72).

TANG, B. (1993). « Orthogonal Array-Based Latin Hypercubes ». In : *Journal of the American Statistical Association* 88.424, pages (cf. p. 70).

— (1998). « Selecting latin hypercubes using correlation criteria ». In : *Statistica Sinica* 8, p. 965–977 (cf. p. 70).

THEOCARIS, P. S. et G. E. STAVROULAKIS (1998). « Multilevel optimal design of composite structures including materials with negative Poisson's ratio ». In : *Structural optimization* 15.1, p. 8–15 (cf. p. 36).

THIÉMARD, É. (2000). « Sur le calcul et la majoration de la discrépance à l'origine ». Thèse de doct. École Polytechnique Fédérale de Lausanne (cf. p. 73).

TOLSMA, J. E. et P. I. BARTON (1998). « On computational differentiation ». In : *Computers & Chemical Engineering* 22.4–5, p. 475–490 (cf. p. 119).

TORCZON, V. (1991). « On the Convergence of the Multidirectional Search Algorithm ». In : *SIAM Journal on Optimization* 1.1, p. 123–145 (cf. p. 20).

TORCZON, V. (1989). « Multi-Directional Search : A Direct Search Algorithm for Parallel Machines ». Thèse de doct. Rice University, Houston,Texas (cf. p. 20).

— (1997). « On the convergence of pattern search algorithms ». In : *SIAM Journal on optimization* 7.1, p. 1–25 (cf. p. 20).

TORN, A. A. (1977). « Cluster Analysis Using Seed Points and Density-Determined Hyperspheres as an Aid to Global Optimization ». In : *Systems, Man and Cybernetics, IEEE Transactions on* 7.8, p. 610–616 (cf. p. 27).

TOROPOV, V. V. et al. (2005). « Design optimization and stochastic analysis based on the moving least squares method ». In : *6th World Congress of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brasil*. 9412, p. 146 (cf. p. 79).

TRELEA, I. C. (2003). « The particle swarm optimization algorithm : convergence analysis and parameter selection ». In : *Information Processing Letters* 85.6, p. 317–325 (cf. p. 29).

TRIKI, E., Y. COLLETTE et P. SIARRY (2005). « A theoretical study on the behavior of simulated annealing leading to a new cooling schedule ». In : *European Journal of Operational Research* 166.1, p. 77–92 (cf. p. 30).

TROVALET, M. (2010). « Sur un modéle micro pour le calcul des structures en composites stratifiés ». Thèse de doct. Cachan : école Normale Supérieure de Cachan (cf. p. 41).

VAN LAARHOVEN, P. J. et E. AARTS (1987). *Simulated annealing : Theory and Applications*. T. 37. Mathematics and Its Applications. Springer Netherlands (cf. p. 30).

Van KEULEN, F., R. T. HAFTKA et N. KIM (2005). « Review of options for structural design sensitivity analysis. Part 1 : Linear systems ». In : *Computer Methods in Applied Mechanics and Engineering* 194.30–33. Structural and Design Optimization, p. 3213–3243 (cf. p. 120).

Van KEULEN, F., R. T. HAFTKA et X.-Y. QU (2000). « Review of options for structural sensitivity calculations ». In : *41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, Atlanta, GA*. AIAA-00-1351 (cf. p. 120).

Van KEULEN, F. et K. VERVENNE (2004). « Gradient-enhanced response surface building ». In : *Structural and Multidisciplinary Optimization* 27.5, p. 337–351 (cf. p. 173).

VAPNIK, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer Verlag, New York (cf. p. 77, 83).

VARADARAJAN, S., W. CHEN et C. J. PELKA (2000). « Robust concept exploration of propulsion systems with enhanced model approximation capabilities ». In : *Engineering Optimization* 32.3, p. 309–334 (cf. p. 89).

VAUCHEZ, P. (1991). « Simulation numérique des processus de mise en forme par la méthode à grand increment de temps ». Thèse de doct. Université Pierre et Marie Curie, 1 vol. (145 f.) (Cf. p. 41).

VAZQUEZ, E. (2005). « Modélisation comportementale de systémes non-linéaires multivariables par méthodes à noyaux et applications ». Thèse de doct. Université Paris XI Orsay (cf. p. 150, 154).

VENTER, G., R. T. HAFTKA et J. H. STARNES (1998). « Construction of Response Surface Approximations for Design Optimization ». In : *AIAA Journal* 36.12, p. 2242–2249 (cf. p. 78).

VIANA, F. A. C. (2013). « Things you wanted to know about the Latin hypercube design and were afraid to ask ». In : *10th World Congress on Structural and Multidisciplinary Optimization, Orlando, Florida, USA* (cf. p. 69).

VIANA, F. A. C. et R. T. HAFTKA (2008). « Using multiple surrogate for metamodeling ». In : *7th ASMO-UK/ISSMO International Conference on Engineering Design Optimization* (cf. p. 88).

VILLANUEVA, D. et al. (2013). « Locating Multiple Candidate Designs with Dynamic Local Surrogates ». In : *10th World Congress on Structural and Multidisciplinary Optimization, Orlando, Florida, USA* (cf. p. 204).

VILLEMONTEIX, J., E. VAZQUEZ et E. WALTER (2009). « An informational approach to the global optimization of expensive-to-evaluate functions ». In : *Journal of Global Optimization* 44.4, p. 509–534 (cf. p. 211).

VIOLEAU, D. (2007). « Une stratégie de calcul pour l'analyse à l'échelle "micro" des endommagements jusqu'à rupture des composites stratifiés ». Thèse de doct. école Normale Supérieure de Cachan (cf. p. 41, 52).

WACKERNAGEL, H. (1995). *Multivariate geostatistics : an introduction with applications*. Springer Verlag (cf. p. 135, 160–161).

WAHBA, G. (1990). *Spline models for observational data*. Society for Industrial Mathematics (cf. p. 151).

WAHBA, G. (1985). « A Comparison of GCV and GML for Choosing the Smoothing Parameter in the Generalized Spline Smoothing Problem ». In : *The Annals of Statistics* 13.4, pages (cf. p. 151).

WALTERS, F. H. et al. (1991). *Sequential simplex optimization : a technique for improving quality and productivity in research, development, and manufacturing*. Chemometrics series. Boca Raton, Fla. : CRC Press (cf. p. 20).

WANG, L., R. GRANDHI et R. CANFIELD (1996). « Multivariate Hermite approximation for design optimization ». In : *International Journal for Numerical Methods in Engineering* 39.5, p. 787–803 (cf. p. 78).

WANG, Y. et K.-T. FANG (1981). « A note on uniform distribution and experimental design ». In : *KeXue TongBao* 26, p. 485–489 (cf. p. 73).

WARNES, J. et B. RIPLEY (1987). « Problems with likelihood estimation of covariance functions of spatial Gaussian processes ». In : *Biometrika* 74.3, p. 640 (cf. p. 150).

WATSON, A. G. et R. J. BARNES (1995). « Infill sampling criteria to locate extremes ». In : *Mathematical Geology* 27.5, p. 589–608 (cf. p. 65).

WELCH, W. J. (1983). « A mean squared error criterion for the design of experiments ». In : *Biometrika* 70.1, p. 205–213 (cf. p. 75).

WENGERT, R. E. (1964). « A simple automatic derivative evaluation program ». In : *Commun. ACM* 7.8, p. 463–464 (cf. p. 126).

WIENER, N. (1938). « The Homogeneous Chaos ». In : *American Journal of Mathematics* 60.4, p. 897–936 (cf. p. 57).

WOLFE, P. (1969). « Convergence Conditions for Ascent Methods ». In : *SIAM Review* 11.2, p. 226–235 (cf. p. 21).

— (1971). « Convergence Conditions for Ascent Methods. II : Some Corrections ». In : *SIAM Review* 13.2, p. 185–188 (cf. p. 21).

WRIGHT, M. H. (1992). « Interior methods for constrained optimization ». In : *Acta Numerica* 1, p. 341–407 (cf. p. 25).

— (1996). « Direct search methods : Once scorned, now respectable ». In : *Numerical Analysis 1995*. Sous la dir. de D. GRIFFITHS et G. WATSON. Addison Wesley Longman (Harlow), p. 191–208 (cf. p. 20).

WRIGHT, S. J. (1997). *Primal-dual interior-point methods*. T. 54. Siam (cf. p. 25).

XUAN, Y. et al. (2009). « Gradient-based Kriging approximate model and its application research to optimization design ». In : *Science in China Series E : Technological Sciences* 52.4, p. 1117–1124 (cf. p. 162).

YAMAZAKI, F. et al. (1988). « Neumann Expansion for Stochastic Finite Element Analysis ». In : *Journal of Engineering Mechanics* 114.8, p. 1335–1354 (cf. p. 55–56).

YAMAZAKI, W., M. P. RUMPFKEIL et D. J. MAVRIPLIS (2010). « Design optimization utilizing gradient/Hessian enhanced surrogate model ». In : *AIAA Paper* 4363, p. 2010 (cf. p. 162).

ZHAO, D. et D. XUE (2010). « A comparative study of metamodeling methods considering sample quality merits ». In : *Structural and Multidisciplinary Optimization* 42.6, p. 923–938 (cf. p. 89–90).

ZIMMERMAN, D. et M. ZIMMERMAN (1991). « A comparison of spatial semivariogram estimators and corresponding ordinary kriging predictors ». In : *Technometrics* 33.1, p. 77–91 (cf. p. 151).

ZIMMERMANN, R. (2013). « On the maximum likelihood training of gradient-enhanced spatial Gaussian processes ». In : *SIAM Journal on Scientific Computing,* in Press (cf. p. 161).

ZONGMIN, W. (1992). « Hermite-Birkhoff interpolation of scattered data by radial basis functions ». In : *Approximation Theory and its Applications* 8.2, p. 1–10 (cf. p. 171).

ZOUTENDIJK, G. (1960). « Methods of feasible directions : A study in linear and non-linear programming ». Thèse de doct. University of Amsterdam (cf. p. 24).