THÈSE

en vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE VERSAILLES ST-QUENTIN-EN-YVELINES

présentée par

Benjamin MARTEAU

Paramétrisation et optimisation sans dérivées pour le problème de calage d'historique

---

JURY

Philippe TOINT	Professeur	Président du Jury
Andrew CONN	Professeur	Rapporteur
Bijan MOHAMMADI	Professeur	Rapporteur
Olivier GOSSELIN	Professeur	Membre du jury
Delphine SINOQUET	Docteur	Membre du Jury
Laurent DUMAS	Professeur	Directeur de Thèse
Didier DING	Professeur	Encadrant

École doctorale :
École doctorale des sciences et techniques de Versailles
Laboratoire et centre de recherche :
Laboratoire de Mathématiques de Versailles & IFP Énergies nouvelles
Directeurs de thèse :
Laurent DUMAS (UVSQ) et Didier Yu DING (IFPEN)

Table des matières

Table des matières **1**

Introduction **5**

1 Paramétrisation du problème de calage d'historique **11**
1.1 État de l'art . . . 11
1.1.1 Introduction . . . 11
1.1.2 Méthodes de déformation graduelle et de déformation de domaine . . . 13
1.1.2.1 Méthode de déformation graduelle . . . 14
1.1.2.2 Méthode de déformation de domaines . . . 15
1.2 Généralisation de la méthode de déformation graduelle . . . 19
1.2.1 Une autre façon d'écrire la déformation de domaines . . . 19
1.2.2 Combinaison de la méthode de déformation graduelle locale avec la méthode de déformation de domaine . . . 20
1.3 Initialisation du problème de calage d'historique : une méthode de patchwork . . . 23
1.3.1 Une méthode de patchwork classique pour initialiser le problème de calage d'historique . . . 23
1.3.2 Amélioration de la méthode de patchwork . . . 24
1.4 Validation numérique des méthodes présentées . . . 26
1.4.1 Description du cas test . . . 26
1.4.2 Initialisation . . . 26
1.4.3 Paramétrisation : Déformation graduelle locale généralisée . . . 29
1.5 Bilan du chapitre . . . 34

2 Optimisation sans dérivées pour le problème de calage d'historique **35**
2.1 État de l'art . . . 35
2.1.1 Méthodes de descente . . . 36
2.1.2 Méthodes de recherche directe . . . 40
2.1.3 Méthodes de type région de confiance . . . 43
2.2 Description du fonctionnement des méthodes de type région de confiance avec interpolation quadratique . . . 45
2.2.1 Un algorithme simpliste . . . 45
2.2.2 Construction des modèles quadratiques . . . 46
2.2.2.1 Décomposition dans une base de polynômes quadratiques . . . 47
2.2.2.2 Calcul des polynômes de Lagrange . . . 47
2.2.3 Λ-unisolvance des ensembles d'interpolation . . . 48
2.2.4 Modèles d'interpolation sous-déterminés . . . 48
2.2.5 Initialisation des modèles quadratiques . . . 50
2.2.6 Précision des modèles et amélioration de la Λ-unisolvance . . . 51

2.2.6.1 Modèles entièrement linéaires . . . 51
2.2.6.2 Lien avec la Λ-unisolvance . . . 52
2.2.6.3 Amélioration de la géométrie des ensembles d'interpolation . . . 53
2.2.7 Un algorithme classique : NEWUOA . . . 53
2.2.8 Auto-correction de la géométrie des ensembles d'interpolation . . . 55
2.3 Une nouvelle méthode de type région de confiance pour les fonctions partiellement séparables : l'algorithme DFO-PSOF . . . 59
2.3.1 Initialisation de l'algorithme DFO-PSOF . . . 60
2.3.2 Une propriété d'auto-correction de la géométrie des ensembles d'interpolation . . . 62
2.3.3 Description de l'algorithme DFO-PSOF . . . 64
2.4 Preuve de convergence de l'algorithme DFO-PSOF . . . 66
2.5 Premiers résultats numériques avec l'algorithme DFO-PSOF . . . 73
2.5.1 Fonctions analytiques partiellement séparables . . . 74
2.5.2 Un cas simple de calage d'historique . . . 75
2.6 Bilan du chapitre . . . 78

3 Résultats numériques sur des modèles de réservoir 79
3.1 Description des deux modèles de réservoir . . . 79
3.1.1 Le cas PUNQ . . . 79
3.1.2 Le cas Brugge . . . 80
3.2 Résultats numériques sur les méthodes de paramétrisation . . . 81
3.2.1 Rappel de la méthodologie . . . 81
3.2.2 Le cas PUNQ . . . 82
3.2.2.1 Initialisation . . . 82
3.2.2.2 Déformation graduelle généralisée . . . 84
3.2.3 Le cas Brugge . . . 87
3.2.3.1 Méthode de Patchwork améliorée . . . 87
3.2.3.2 Déformation graduelle généralisée . . . 90
3.2.3.3 Importance du prétraitement . . . 93
3.2.3.4 Calage d'historique obtenu . . . 94
3.2.3.5 Seconde optimisation . . . 95
3.2.3.6 Discussions sur le cas Brugge . . . 97
3.3 Résultats numériques d'optimisation . . . 98
3.3.1 Cas PUNQ . . . 98
3.3.1.1 Séparation des variables . . . 98
3.3.1.2 Tests numériques . . . 98
3.3.1.3 Calage d'historique obtenu . . . 100
3.3.2 Le cas Brugge . . . 105
3.3.2.1 Séparation des variables . . . 105
3.3.2.2 Tests numériques . . . 105
3.3.2.3 Seconde optimisation . . . 108
3.3.2.4 Calage d'historique obtenu . . . 109
3.4 Bilan du chapitre . . . 114

Conclusion 115

Annexe 117

Table des figures 127

Introduction

A. Optimisation en ingénierie pétrolière

Les environnements de travail entourant la récupération pétrolière deviennent de plus en plus complexes et délicats. Les nouveaux champs découverts se trouvent la plupart du temps dans des conditions extrêmes, tandis que le pourcentage de récupération des champs déjà exploités devient très difficile à améliorer. Dans de telles conditions et en l'absence de techniques d'optimisation de production, il serait difficile de justifier économiquement l'investissement dans la production de champs existants : le besoin en modèles de prédiction de la production précis et fiables devient évident. Ces dernières années, l'explosion des puissances de calcul informatique a rendu possible le développement de tels modèles. Il est désormais possible de simuler en un temps raisonnable (quelques heures) les équations complexes qui régissent les écoulements des fluides en milieu poreux. L'enjeu est alors d'ajuster au mieux les paramètres du modèle de façon à ce que le résultat de ces simulations représente au mieux la réalité.

Le fait de pouvoir simuler numériquement le comportement d'un champ pétrolier permet de tester de nombreuses configurations de systèmes de production et a amené naturellement l'industrie pétrolière à utiliser des techniques d'optimisation. Des problèmes de natures très variables ont alors pu être traités numériquement en amont de l'exploitation. On trouve par exemple dans la littérature des façons d'optimiser le placement d'un ou plusieurs nouveaux puits pour augmenter la récupération d'un réservoir [5, 7, 13, 51, 79] ou des techniques pour maximiser la production d'un champ déjà exploité en ajustant entre autres les pressions ou débits aux différents puits [31, 32, 73, 109, 114]. Un point commun à tous ces problèmes est qu'ils nécessitent un modèle de réservoir suffisamment prédictif pour pouvoir être significatif. Parmi les grandes incertitudes que l'on peut avoir sur les modèles de réservoir se trouvent la distribution de faciès géologiques ainsi que leurs propriétés pétrophysiques comme la perméabilité, la porosité, etc... Le domaine de la géostatistique permet à partir de mesures physiques de données statiques ou de données sismiques 3D de construire un modèle en extrapolant les propriétés mesurées aux nombreuses mailles du modèle [35, 89]. Malheureusement, les mesures ne sont que très parcellaires tout en ayant toujours de grandes incertitudes et erreurs d'interprétation, elles sont donc insuffisantes pour déterminer un réservoir dans son ensemble. Une idée pour affiner le modèle est d'exploiter nos connaissances sur données dynamiques comme l'historique de production ou les mesures sismiques 4D pour améliorer la précision du modèle (ou simplement réduire ses incertitudes). On cherche à faire en sorte que lors des simulations d'écoulement dans le modèle de réservoir, les données dynamiques soient le plus proche possible des données mesurées. L'approche la plus couramment utilisée dans l'industrie consiste à faire appel au jugement d'un expert pour choisir à la main ces paramètres via un processus d'essai/erreur. Cependant le calage d'historique puits par puits des données de production est généralement un processus long et fastidieux qui nécessite beaucoup d'efforts et d'expertise de la part des ingénieurs de réservoir. De plus, modifier à la main certains paramètres peut altérer la continuité géologique du modèle,

réduisant par la même occasion une grande partie de son pouvoir prédictif. On cherche donc une paramétrisation efficace du modèle permettant de conserver son intégrité ainsi qu'une manière automatique d'optimiser simultanément leurs valeurs : c'est le calage d'historique assisté.

B. Le calage d'historique

(i) Motivations

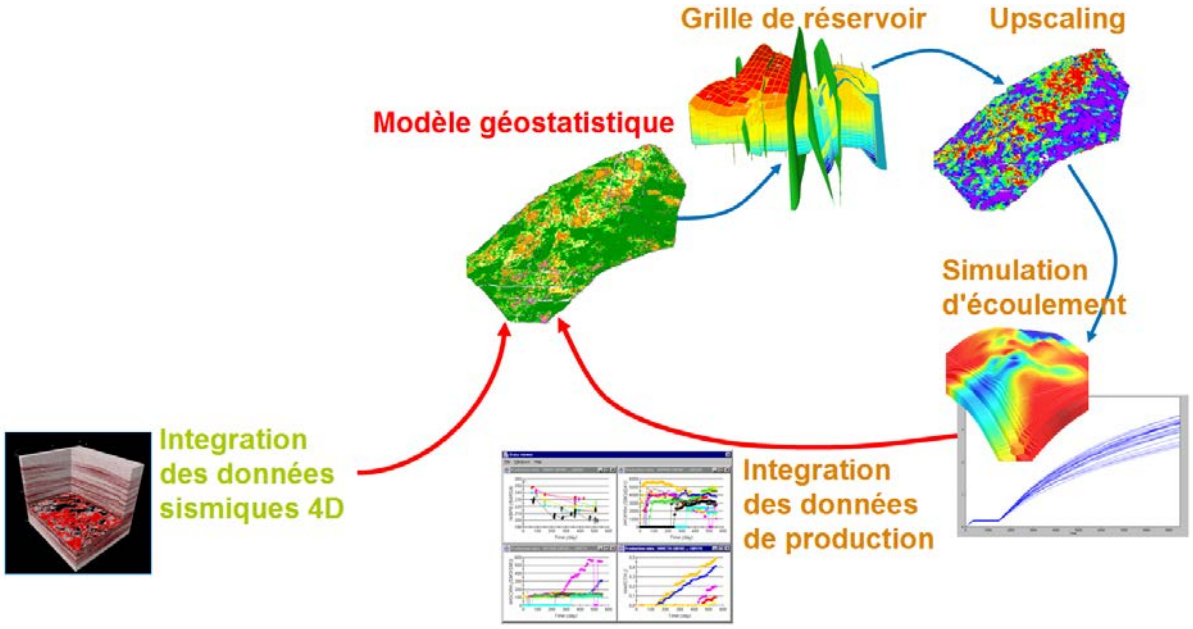
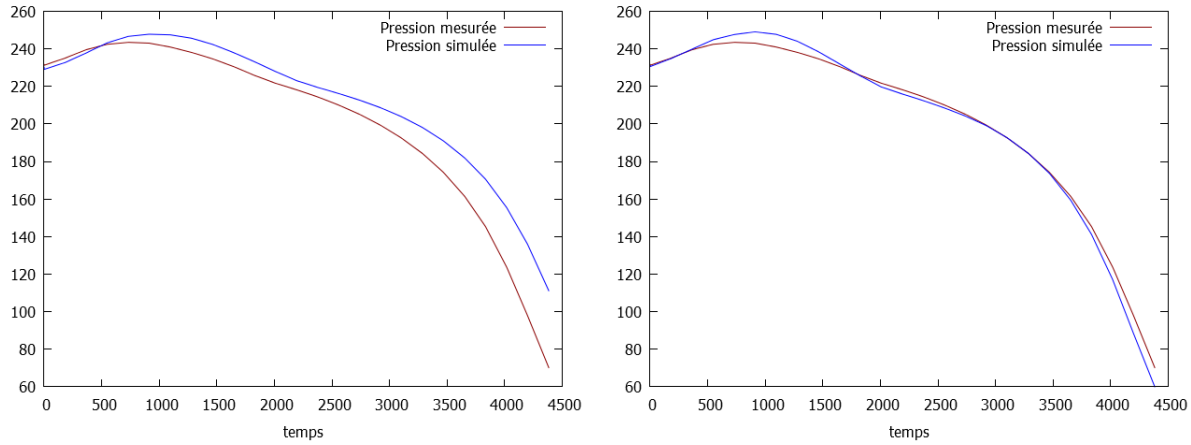


FIGURE 1 – Boucle illustrant le calibrage des paramètres d'un modèle par calage d'historique.

Le calage d'historique revient à contraindre les paramètres du modèle de réservoir de façon à réduire ses incertitudes. C'est un processus illustré par la boucle présentée en Figure 1 qui permet d'obtenir des simulations plus prédictives du comportement du réservoir. On peut décrire cette boucle comme suit : un modèle géostatistique est créé à partir de mesures physiques (sismique, mesures ponctuelles aux puits, etc.) sur un maillage fin avant de subir un processus d'upscaling qui rend possible les simulations d'écoulement en un temps raisonnable. On simule alors les écoulements dans le réservoir de façon à pouvoir comparer l'historique de production simulé avec les données mesurées. Enfin, dans la dernière partie de la boucle, on fait varier les paramètres du modèle de façon à faire en sorte que les historiques de production ou de sismique 4D simulés se rapprochent des mesures réelles. Le processus de calage d'historique revient à itérer toutes les étapes de cette boucle jusqu'à trouver les valeurs optimales des paramètres du modèle. Une comparaison entre les historiques mesurés et simulés de la pression au fond du deuxième puits producteur du réservoir Brugge [81, 82] est présenté en Figure 2 : avant calage d'historique à gauche et après calage à droite. Le calage d'historique permet ici de réduire de beaucoup les erreurs commises par le modèle de réservoir.

On peut caler toutes sortes de données via calage d'historique, même si ce sont souvent les productions d'huile et d'eau que l'on désire prédire au mieux. De la même façon, les paramètres dont dépend le modèle et sur lesquels on peut agir pour améliorer le calage peuvent être très nombreux et de natures radicalement différentes : on trouve par exemple de nombreuses grandeurs physiques comme les distributions de perméabilité ou de porosité, la perméabilité relative



(a) avant calage d'historique

(b) après calage d'historique

FIGURE 2 – Calage de la pression au fond d'un puits producteur du réservoir Brugge avant et après calage d'historique.

(k_r), la pression capillaire (p_c), la viscosité de l'huile, la présence d'aquifère ou de failles, etc. Pour caler les distributions de perméabilité et de porosité générées par un modèle géostatistique, il est nécessaire de paramétrer ce modèle de façon à conserver ses propriétés géostatistiques. Les méthodes de déformation graduelle [92] ou de déformation de domaines [37] sont de bons exemples de techniques pour y parvenir. Devant cette multitude de paramètres et de données à caler, le coût en temps de calcul de l'optimisation pour le calage d'historique peut très vite devenir très élevé. Il est donc nécessaire de développer des techniques de sélection des paramètres et d'optimisation efficaces adaptées au problème.

(ii) Formulation du problème

Notons F la fonction d'erreur entre les données simulées et les données mesurées et $x = (x_1, \ldots, x_p)$ les paramètres dont dépend le modèle de réservoir :

$$F(x_1,\ldots,x_p) = \frac{1}{2}\sum_{i=1}^{n_1}\left[\frac{\omega_i^P}{N_P(i)}\sum_{j=1}^{N_P(i)}\left(\frac{P_{i,j}^{obs}(x)-P_{i,j}^{sim}(x)}{\sigma_{i,j}^P}\right)^2\right] + \frac{1}{2}\sum_{i=1}^{n_2}\left[\frac{\omega_i^S}{N_S(i)}\sum_{j=1}^{N_S(i)}\left(\frac{S_{i,j}^{obs}(x)-S_{i,j}^{sim}(x)}{\sigma_{i,j}^S}\right)^2\right] \quad (1)$$

où on considère :

- n_1 : nombre de puits,
- n_2 : nombre de séries de mesures sismiques effectuées,
- $N_P(i)$: nombre temps de mesures effectuées au puits i,
- $N_S(i)$: nombre de données sismiques observées pour la série de données i,
- $P_{i,j}^{obs}$: donnée de production observée au puits i et au temps associé j,
- $P_{i,j}^{sim}$: donnée de production simulée au puits i et au temps associé j,
- $\sigma_{i,j}^P$: écart type des données de production,
- $\sigma_{i,j}^S$: écart type des données sismiques,
- ω_i^P et ω_i^S : poids associés respectivement aux données de production et aux données sismiques .

L'expression de la fonction F est simplement une formulation aux moindres carrés de l'erreur entre les données mesurées et les données observées. On cherchera dans une optique de calage d'historique à minimiser cette erreur.

Pour simplifier l'écriture, on réécrit F comme suit :

$$F(x_1, \ldots, x_p) = \sum_{i=1}^{n} f_i(x_1, \ldots, x_p)$$

où $n = n_1 + n_2$. Le calage d'historique revient donc à résoudre un problème d'optimisation non contraint de la forme :

$$\begin{cases} \text{Trouver } (x_1^*, ..., x_p^*) = arg \min\limits_{x \in \mathbb{R}^p} F(x_1, \ldots, x_p) \\ \\ \quad \text{avec } F(x_1, ..., x_p) = \sum\limits_{i=1}^{n} f_i(x_1, \ldots, x_p) \end{cases}$$

(iii) Caractéristiques du problème

Coût de calcul de la fonction objectif. L'une des caractéristiques principales du problème de calage d'historique est le coût d'évaluation de la fonction objectif. En effet, celle-ci nécessite une simulation complète d'un simulateur d'écoulement qui est un problème lui même très complexe : la multitude des phénomènes physiques considérés (écoulements multiphasiques en milieux poreux, fracturation du domaine, connectivité de canaux d'écoulement privilégiés, etc.) rendent ce calcul très coûteux. Seul un nombre restreint d'évaluations peut donc être effectué pour obtenir un résultat en temps raisonnable. Il est donc primordial d'avoir une technique d'optimisation adaptée ne demandant qu'un petit nombre d'évaluations pour arriver à une convergence acceptable.

Nombre de paramètres. On a vu plus haut que la fonction objectif F dépend potentiellement d'un nombre très élevé de paramètres du fait de la nature complexe des simulations d'écoulements et des grandes incertitudes sur le modèle amenées par le relativement faible nombre de mesures. Une augmentation du nombre de paramètres dans un algorithme d'optimisation en augmente radicalement le coût. Un des gros enjeux du calage d'historique consiste donc à introduire un nombre restreint de paramètres influents tout en maintenant un nombre suffisant pour pouvoir caler au mieux les données de production.

Absence des dérivées de la fonction objectif. La plupart des simulateurs industriels ne disposent pas de problème adjoint. Additionellement, le problème considéré est bruyant, il est donc difficile d'estimer les dérivées de la fonction objectif en un temps raisonable. Trouver une méthode d'optimisation efficace ne nécessitant aucune estimation des dérivées de la fonction objectif est alors un enjeu majeur pour le calage d'historique.

Séparabilité partielle de la fonction objectif. Une bonne approximation pour la modélisation du champ pétrolier est de considérer que la production d'un puits n'est affectée que par un nombre restreint de paramètres. Certains paramètres agissent par exemple localement sur une région de réservoir (skin d'un puits, paramètre de déformation graduelle locale, etc.). Il est donc raisonnable de penser que de tels paramètres n'influenceront que très peu les parties du modèle qui sont éloignées de la région qui leur est associée.

On peut illustrer facilement ce concept sur la Figure 3. Si par exemple certains paramètres ne modifient pas les propriétés du résevoir dans la zone circulaire centrée autour du puits producteur

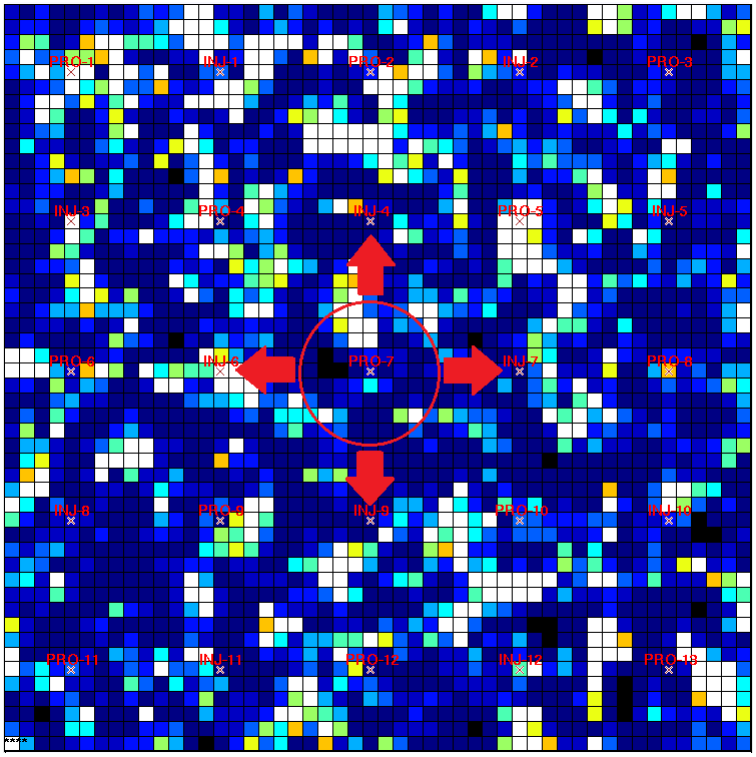


FIGURE 3 – Illustration de la séparabilité partielle de la fonction objectif associée à champ synthétique.

PRO-7, il est légitime de considérer que les seules données de production qui seront affectées sont celles du puits PRO-7 et des quatre puits injecteurs les plus proches, INJ-4, INJ-6, INJ-7 et INJ-9. Faire cette approximation revient à dire que la fonction objectif est partiellement séparable. On dit qu'une fonction est partiellement séparable lorsqu'elle peut s'écrire sous la forme :

$$F(x_1, \ldots, x_p) = \sum_{i=0}^{n} f_i(x_{i,1}, \ldots, x_{i,p_i}) \text{ avec } \forall i, p_i < p \tag{2}$$

C'est cette des propriété majeure du problème que l'on va essayer d'exploiter de façon à réduire considérablement le nombre d'évaluations nécessaires à l'optimisation.

C. Description des chapitres

On se focalise ici sur deux aspects principaux du problème de calage d'historique :

- la définition de paramètres influents déformant continument le modèle de réservoir,
- le développement d'une méthode d'optimisation sans dérivées n'utilisant qu'un nombre restreint de simulation d'écoulements pour arriver à convergence.

On introduit dans le chapitre 1 une nouvelle méthode de paramétrisation généralisant la méthode de déformation graduelle locale [46, 54]. Cette méthode permet de combiner deux réalisations de modèle ou plus à l'intérieur de zones paramétrées. On laisse alors à l'algorithme d'optimisation la possibilité d'ajuster à la fois les paramètres de déformation graduelle et la forme des zones. On remédie ainsi à l'un des problèmes majeurs de la déformation graduelle locale : la dépendance à la forme initiale des zones de la réalisation obtenue en fin d'optimisation. Un autre

avantage de cette méthode de paramétrisation réside dans le fait qu'il est plutôt facile de trouver les indépendances entre les paramètres introduits et les données de production des différents puits, déterminant ainsi la séparabilité partielle de la fonction objectif. Un test numérique sur un modèle de réservoir simple est présenté en fin de chapitre, permettant de valider à la fois la méthode elle-même et le code développé.

Dans le chapitre 2, on présente une nouvelle méthode d'optimisation sans dérivées adaptée aux fonctions partiellement séparables. Il s'agit d'une méthode de type région de confiance fondée sur des modèles d'interpolation quadratiques locaux de la fonction objectif. L'adaptation à la séparabilité partielle de la fonction objectif se fait au niveau de la construction des modèles : au lieu de construire un modèle pour la fonction objectif globale, la méthode construit un modèle pour chaque fonction objectif locale (les f_i dans l'équation (2)). Chacune de ces fonctions objectif locales ne dépendant que d'un nombre très restreint de paramètres, les modèles d'interpolation peuvent être construits avec seulement un petit nombre de points sans détériorer leur précision. Cette nouvelle méthode permet alors une convergence beaucoup plus rapide lorsque la fonction objectif est partiellement séparable. Des premiers tests numériques sont présentés dans ce chapitre sur des fonctions analytiques ainsi que sur un cas simple de réservoir. À noter que la convergence théorique de la méthode vers un point critique est par ailleurs démontrée dans ce chapitre.

Le chapitre 3 est quant à lui consacré à des tests des méthodes présentées précédemment sur des modèles de réservoir plus complexes : les cas PUNQ et Brugge, deux réservoirs synthétiques couramment utilisés dans la littérature pour valider les nouvelles méthodes numériques. Plusieurs variantes de la méthode de paramétrisation présentée sont d'abord testés au chapitre 1 puis, avec les paramètres ainsi définis, la nouvelle méthode d'optimisation est appliquée avec des résultats très encourageants pour de futures problèmes impliquant des réservoirs réels.

D. Publication & articles

Les travaux réalisés au cours de cette thèse ont conduit à la publication ou la soumission de plusieurs articles et proceedings :

- "A generalization of the local gradual deformation method using domain parameterization", publié dans *Computers and Geosciences* [69],
- "A generalized local gradual deformation method", publié dans *Proceedings of the 15th annual conference of the international association for mathematical geosciences* [68],
- "Derivative free optimization for partially separable functions", soumis à *Optimization Methods and Software*,
- "A derivative free optimization method adapted to partially separable functions for history matching problems", soumis à l'édition spéciale ECMOR du journal *Computational Geosciences*.

Chapitre 1

Paramétrisation du problème de calage d'historique

Dans ce chapitre, on introduit une nouvelle méthode de paramétrisation géostatistique ainsi qu'une façon efficace d'initialiser les paramètres définis. La méthode proposée est fondée sur les principes des méthodes de déformation graduelle locale [92] et de déformation de domaines [37]. La déformation graduelle locale consiste à combiner plusieurs réalisations de modèle à l'intérieur de zones préalablement définies sur le réservoir tandis que la déformation de domaines revient à choisir une réalisation de modèle à l'intérieur de ces zones, une autre à l'extérieur et à modifier la réalisation de modèle en faisant varier la forme des zones. Notre méthode combine les avantages des deux méthodes en permettant de modifier la forme des zones dans lesquelles la déformation graduelle locale est effectuée. Le fait de pouvoir faire varier la forme des zones supprime l'un des facteurs limitant principaux de la méthode de déformation graduelle locale : la réalisation de modèle obtenue en fin d'optimisation avec notre méthode n'est plus que marginalement dépendante de la forme des zones initialement définies. De plus, la flexibilité apportée par notre méthode dans le choix des zones nous a permis de développer une méthodologie d'initialisation de type "patchwork" qui nous permet de garantir le choix d'un bon point initial.

On présente en section 1.1 un état de l'art des méthodes de paramétrisation existantes ainsi qu'une description plus précise des méthodes de déformation graduelle et de déformation de domaines. La nouvelle méthode de paramétrisation est ensuite décrite en section 1.2 tandis que la méthodologie développée pour l'initialiser est présentée en section 1.3. Enfin, une première validation numérique des méthodes proposées sur un cas simple de réservoir est donnée en section 1.4.

1.1 État de l'art

1.1.1 Introduction

(i) Paramétrisation

Un réservoir pétrolier dans son ensemble comprend de nombreuses hétérogénéités dues à des phénomènes physiques ou géologiques variés et ce, à différentes échelles. Les phénomènes d'accrétion et de sédimentation peuvent être par exemple responsables de l'apparition de grosses structures géologiques à l'intérieur du réservoir comme des chenaux qui, à leur tour, contiennent des hétérogénéités à une échelle plus petite. Rendre compte précisément d'une telle complexité dans un modèle contenant des millions de mailles nécessite l'ajustement d'un très grand nombre de paramètres. Ceci est généralement infaisable dans la pratique du fait du faible nombre d'observations à disposition. Il est donc nécessaire de s'appuyer sur des techniques de paramétrisation

permettant de réduire le nombre de paramètres tout en conservant l'intégrité du modèle. On peut distinguer trois grandes familles de paramétrisation :

— une famille contenant les méthodes apparentées à des techniques d'expansion
— une famille dont les méthodes sont fondées sur une représentation de la variabilité spatiale par blocs
— une famille basée sur des concepts géostatistiques.

La première famille d'approches comprend trois méthodes principales : la méthode d'expansion de Karhunen-Loève [45, 77, 93] ou analyse linéaire en composante principale [97, 59], la transformée en cosinus discrète [56] et la transformée en ondelette [94, 95]. Ces trois méthodes ont pour but de réduire le nombre de paramètres total en travaillant sur des espaces de dimension plus faible que celui du problème général.

La méthode de Karhunen-Loève consiste à identifier les valeurs propres de la matrice de covariance du modèle géostatistique et à négliger le sous-espace généré par les vecteurs propres associés aux plus faibles valeurs propres. Il est alors possible de travailler sur un espace de dimension plus faible tout en conservant la majeure partie de l'information sur les variabilités spatiales du réservoir.

Dans la méthode utilisant la transformée en cosinus discrète, les propriétés du réservoir sont modélisées à l'aide d'une transformée en cosinus discrète, c'est-à-dire comme une série de fonctions cosinus. Cette somme est ensuite ordonnée suivant les coefficients de la transformée et les termes correspondant à des valeurs trop faibles sont négligés. La dimension de l'espace sur lequel les calculs sont réalisés est donc réduite et le nombre de paramètres s'en trouve limité.

Enfin, la dernière méthode consiste à effectuer une transformée en ondelette des propriétés pétrophysiques du réservoir puis à sélectionner les termes souhaités en fonction de leur sensibilité aux données de production. Le modèle résultant est à nouveau de dimension réduite et permet donc de diminuer le nombre total de paramètres. Dans chacune de ces méthodes, on réduit la dimension de l'espace de travail en identifiant puis en négligeant un sous-espace moins influent. Il est donc possible de détériorer fortement les propriétés pétrophysiques du modèle de réservoir si on néglige une partie trop importante de l'espace.

La deuxième famille de méthodes consiste à regrouper les mailles de domaines en blocs et à faire varier les moyennes des propriétés pétrophysiques dans chaque bloc. La méthode de zonation [55, 104] a souvent été utilisée pour le calage d'historique. Dans cette méthode, les perméabilités et porosités des mailles sont considérées constantes dans chacun des blocs préalablement définis. Cette méthode permet bien une diminution du nombre total de paramètres mais ne rend pas correctement compte de la variabilité spatiale des propriétés du réservoir. La méthode de *gradzone* [10] est une amélioration de la méthode de zonation dans le sens où elle permet des modifications à l'intérieur des blocs pendant le processus de calage. Cependant, elle ne permet pas non plus de conserver les informations statiques de variabilité spatiale.

Les travaux réalisés au cours de la thèse portent principalement sur la dernière famille de méthodes dont l'avantage est de permettre la conservation de la structure et de la variabilité géologique du modèle. Quelques méthodes sont à citer dans cette famille : la méthode des points pilotes [67, 88], la méthode de déformation graduelle [53, 54, 92], la méthode de déformation de domaines (Ding et Roggero [37]) et la méthode de la perturbation des probabilités (Hoffman et Caers [52]). Le fait que ces méthodes soient fondées sur des propriétés géostatistiques fait qu'elles partagent plusieurs propriétés importantes. Elles modifient en effet toutes les propriétés pétrophysiques du modèle de façon globale ou locale tout en conservant les propriétés de variabilité spatiale intactes. De plus, elles permettent aussi de réduire de façon importante le nombre

de paramètres à prendre en compte dans le calage d'historique. La méthode des points pilotes par exemple consiste à sélectionner quelques points dans le modèle de réservoir et à ajuster les propriétés pétrophysiques qui leur sont attribuées pour réduire la fonction objectif. Les valeurs en tous les autres points du réservoir sont alors modifiées afin de conserver les propriétés de variabilité spatiale du modèle. Les méthodes de déformation graduelle et de domaines reviennent quant à elles à considérer des combinaisons linéaires de plusieurs réalisations du modèle géostatistique et à définir en tant que paramètres les coefficients de ces combinaisons. Ce sont ces deux méthodes que l'on a tenté de généraliser, elles seront donc décrites plus en détail dans la section suivante.

(ii) Initialisation
Un autre aspect du problème de calage d'historique qui peut influencer fortement le résultat de l'optimisation est son initialisation. L'une des difficultés des méthodes de paramétrisation géostatistiques locales est la définition des zones dans lesquelles elles agissent : si ces zones d'influence ne sont pas choisies de façon pertinente, la diminution de la fonction objectif ne pourra être que très limitée. D'un autre côté, le choix du vecteur initial de paramètres, selon son éloignement par rapport au point initial, peut jouer énormément sur le nombre d'évaluations de la fonction objectif nécessaires pour arriver à convergence de l'algorithme d'optimisation utilisé. On a donc besoin d'une méthode efficace et pertinente pour initialiser le problème. On peut trouver dans la littérature une méthode dite de patchwork [90] consistant à diviser le modèle de réservoir en zones délimitées par des mailles puis à sélectionner pour chacune de ces zones une réalisation de modèle. Cette méthode peut donner de bons résultats sur la valeur de la fonction objectif mais n'offre aucune garantie : il est tout à fait possible d'obtenir un point initial moins bon avec cette méthode. Par conséquent, il est courant de débuter l'optimisation de calage d'historique avec la meilleur réalisation de modèle générée initialement [11, 18].

Dans ce chapitre, on s'intéresse plus spécifiquement aux méthodes de déformation graduelle et de déformation de domaines. Une revue rapide de ces deux méthodes est présentée en section 1.1.2. La méthode de paramétrisation que nous avons développée généralise la méthode de déformation graduelle locale et est décrite en section 1.2. Elle permet généralement une plus grande souplesse dans le choix des zones et des paramètres associés comparativement aux méthodes existantes. La section 1.3.2 se focalise quant à elle sur l'initialisation de cette méthode de paramétrisation. Enfin, la section 1.4 montre un premier résultat numérique simple permettant de valider nos méthodes.

1.1.2 Méthodes de déformation graduelle et de déformation de domaine

Le principe des méthodes de déformation graduelle et de déformation de domaines est fondé sur la façon dont sont générés les modèles géostatistiques. Le peu de données disponibles implique que les modèles se doivent d'extrapoler dans les parties inconnues du réservoir. Les méthodes de génération de modèle proposent ainsi des modèles qui vérifient des propriétés de variabilité spatiale statistiques. Certaines méthodes comme FFT-MA (Fast Fourier Transform Moving Average) [89] permettent de générer des informations structurelles (moyenne, variance, longueur de corrélation, etc.) sur des bruits blancs décorrélés. On génère d'abord un bruit blanc Gaussien qui est ensuite manipulé pour faire en sorte qu'il vérifie les propriétés statistiques souhaitées. Une réalisation d'un modèle est alors liée à ce tirage aléatoire de bruit blanc Gaussien par une fonction G telle que

$$M = G(Z)$$

Où M est la réalisation du modèle associée au bruit blanc Gaussien Z.

1.1.2.1 Méthode de déformation graduelle

La propriété simple suivante est à la base des méthodes de paramétrisation qui nous intéressent.

Proposition 1. *Soient* $(Z_i)_{i=1..N}$ *des variables aléatoires indépendantes de loi normale* $\mathcal{N}(\mu_i, \sigma_i^2)_{i=1..N}$ *et* $(a_i)_{i=1..N}$ *des réels, alors :*

$$Z = \sum_{i=1}^{N} a_i Z_i$$

suit aussi une loi normale $\mathcal{N}(\sum_i a_i\mu_i, \sum_i a_i^2\sigma_i^2)$

La propriété 1 implique que, pour des Z_i suivant une loi normale centrée réduite, si l'on choisit les a_i tels que $\sum_i a_i^2 = 1$, la combinaison $Z = \sum_i a_i Z_i$ suit elle aussi une loi normale centrée réduite. Ainsi, à partir de N réalisations indépendantes du modèle géostatistique, on peut, grâce à une combinaison linéaire des bruits blancs Gaussiens, en obtenir un nouveau. La méthode de déformation graduelle consiste à faire dépendre les a_i d'un paramètre ρ^1 tout en s'assurant que quelle que soit la valeur de ce paramètre, $\sum_i a_i(\rho^1)^2 = 1$.
Par exemple, pour la combinaison de 2 réalisations on choisit :

$$Z = \cos(\rho^1)Z_1 + \sin(\rho^1)Z_2 \tag{1.1}$$

La Figure 1.1 illustre la façon de construire un nouveau bruit blanc Gaussien standard à partir de deux bruits blancs existants.

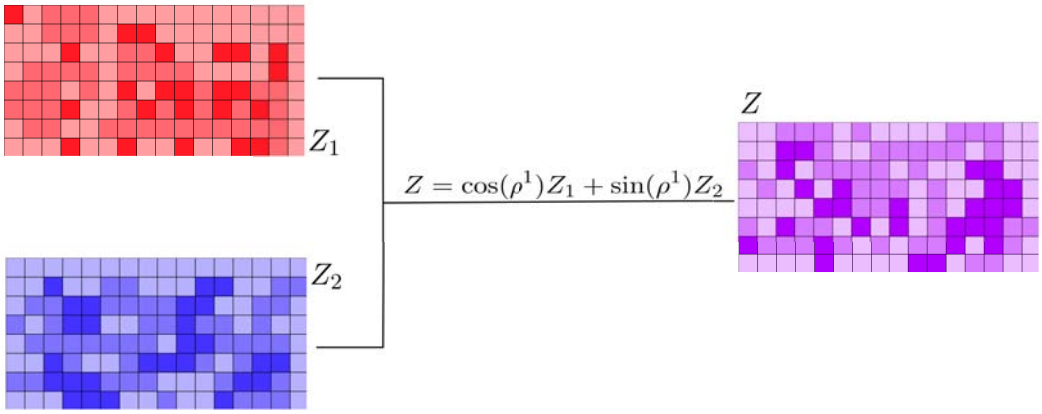


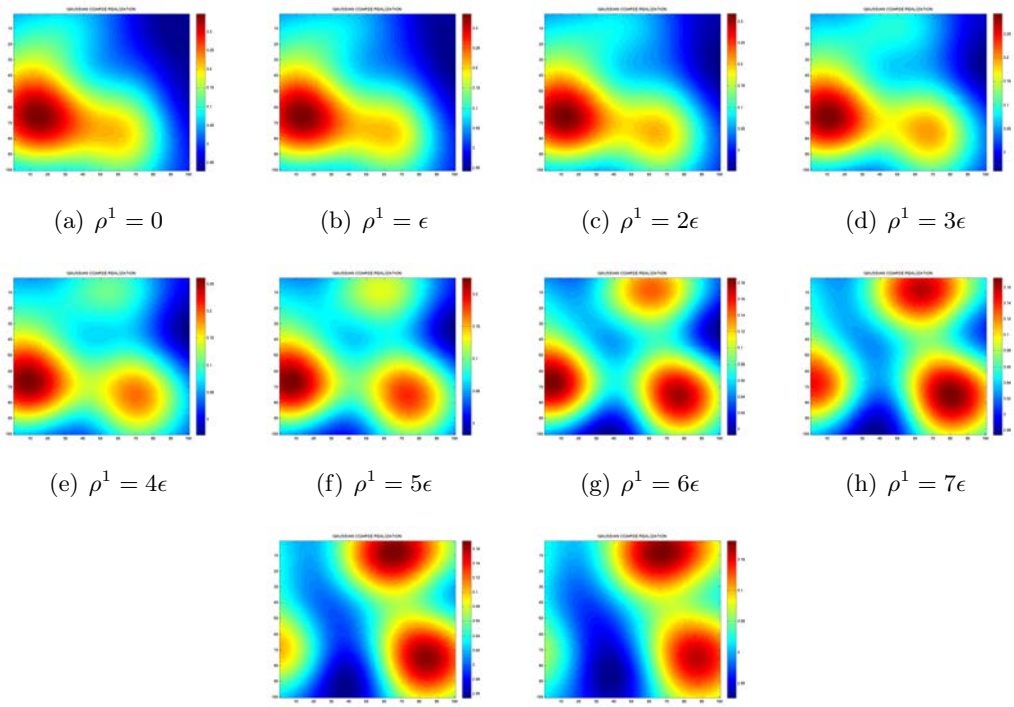
FIGURE 1.1 – Combinaison de deux bruits blancs Gaussiens par déformation graduelle.

Une variation continue du paramètre ρ^1 entraîne une variation continue du modèle, ce qui permet de lancer simplement un processus d'optimisation sur ce paramètre. La Figure 1.2 illustre la modification d'un modèle avec un paramètre de déformation graduelle.
Pour combiner $N+1$ bruits blancs Gaussiens $(Z_0, \ldots, Z_N)$, on introduit N paramètres $\rho = (\rho^1, \ldots, \rho^N)$. La méthode de déformation graduelle est donnée par la formule suivante qui permet d'assurer que $\sum_i a_i(\rho)^2 = 1$:

$$Z = \prod_{i=1}^{N} \cos(\rho^i)Z_0 + \sum_{i=1}^{N-1} \sin(\rho^i) \prod_{k=i+1}^{N} \cos(\rho^k)Z_i + \sin(\rho^N)Z_N \tag{1.2}$$

Il est toujours facile de vérifier qu'avec cette formulation, $\forall \rho \in \mathcal{R}^N$, $\sum_{i=1}^{N} a_i^2(\rho) = 1$.

Par ailleurs, il est possible de n'appliquer cette méthode que localement. Il est aisi possible de grouper les mailles du modèle par zones et de ne modifier les bruit blancs Gaussiens que sur



(a) $\rho^1 = 0$ (b) $\rho^1 = \epsilon$ (c) $\rho^1 = 2\epsilon$ (d) $\rho^1 = 3\epsilon$

(e) $\rho^1 = 4\epsilon$ (f) $\rho^1 = 5\epsilon$ (g) $\rho^1 = 6\epsilon$ (h) $\rho^1 = 7\epsilon$

(i) $\rho^1 = 8\epsilon$ (j) $\rho^1 = 9\epsilon$

FIGURE 1.2 – variations d'un modèle avec un paramètre de déformation graduelle ; de (a) à (j), ρ^1 varie d'un pas constant ϵ.

ces domaines. En clair, si l'on divise le modèle en 2 zones $(zones_1, zones_2)$ et que l'on dispose de 3 bruits blancs Gaussiens standards (Z_0, Z_1, Z_2) :

$$Z_0 = \begin{bmatrix} Z_{0,zone_1} \\ Z_{0,zone_2} \end{bmatrix}, Z_1 = \begin{bmatrix} Z_{1,zone_1} \\ Z_{1,zone_2} \end{bmatrix} \text{ et } Z_2 = \begin{bmatrix} Z_{2,zone_1} \\ Z_{2,zone_2} \end{bmatrix} \tag{1.3}$$

Z_0 est appelée la réalisation de base. Si Z est défini par :

$$Z = \begin{bmatrix} \cos(\rho_1^i) Z_{0,zone_1} + \sin(\rho_1^i) Z_{1,zone_1} \\ \cos(\rho_2^i) Z_{0,zone_2} + \sin(\rho_2^i) Z_{2,zone_2} \end{bmatrix} \tag{1.4}$$

alors Z est toujours un bruit blanc et peut donc légitimement être utilisé pour générer une réalisation du modèle. Les combinaisons effectuées sont illustrées sur la Figure 1.3.

Dans ce cas, on combine la réalisation de base Z_0 avec Z_1 dans la première zone et avec Z_2 dans la deuxième. L'introduction de deux paramètres de déformation graduelle permet de gérer indépendamment deux parties du modèle et autorise une plus grande souplesse dans le choix des réalisations à combiner.

1.1.2.2 Méthode de déformation de domaines

La méthode de déformation de domaines présente des similarités avec la méthode de déformation graduelle locale. Le modèle de réservoir est divisé en différentes zones auxquelles sont associées un bruit blanc Gaussien standard unique. La réalisation du modèle est alors modifiée à travers la déformation de la forme des zones. La Figure 1.4 présente un exemple de modèle de réservoir dans lequel une zone a été créée. Sur cet exemple, la réalisation de modèle est construite

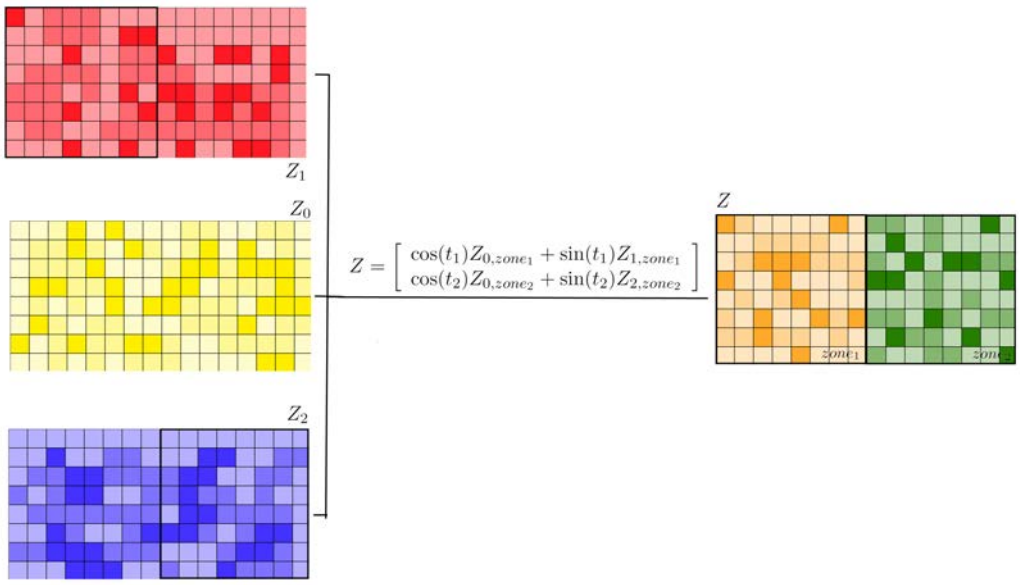


FIGURE 1.3 – Combinaison locale de deux bruits blancs Gaussiens par déformation graduelle.

avec le bruit blanc Gaussien Z_1 à l'intérieur du domaine $R1$ et le bruit blanc Gaussien Z_0 à l'extérieur. Il est clair sur cette figure que la réalisation de modèle construite est largement dépendante de la forme de la zone R_1 et qu'une paramétrisation de cette forme est bien une paramétrisation du modèle. Cependant, la valeur aléatoire utilisée pour construire la réalisation

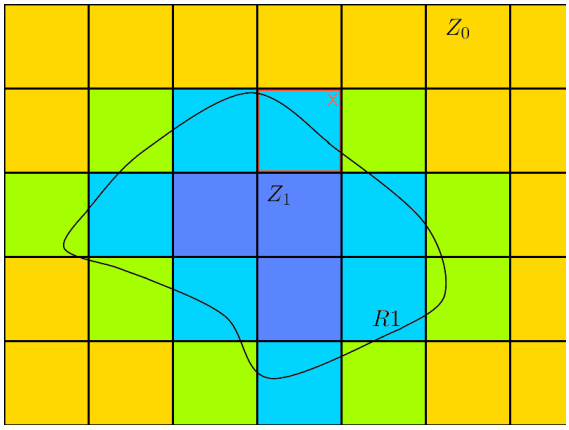


FIGURE 1.4 – Modèle de réservoir dans lequel le bruit blanc Z_1 est appliqué dans R_1 et Z_0 hors de R_1.

de modèle n'est pas clairement définie sur les mailles partiellement à l'intérieur d'une ou plusieurs zones. La maille X par exemple n'est pas entièrement à l'intérieur de R_1. On pourrait choisir pour cette maille l'une ou l'autre des réalisations en fonction de la zone qui contient la plus grande part de la maille, mais ce choix présente deux inconvénients : la forme du domaine se trouve très limitée par la forme des mailles, et la continuité du modèle avec la variation des paramètres peut être difficile à assurer. On choisit donc pour cette maille une réalisation générée par un bruit blanc issu d'une combinaison linéaire de Z_0 et Z_1,

$$Z_X = a_1 Z_{0,X} + a_2 Z_{1,X} \tag{1.5}$$

où les a_i dépendent des paramètres définissant la forme des domaines.
Comme pour la déformation graduelle, pour que Z_X puisse toujours générer une réalisation du modèle, il faut que les coefficients de la combinaison vérifient $a_1^2+a_2^2 = 1$. Une manière de prendre en compte la forme du domaine est de rendre ces coefficients proportionnels à la proportion de la maille dans chaque domaine :

$$a_i = \frac{vol(X \cap R_i)}{\sqrt{vol^2(X \cap R_1) + vol^2(X \cap R_2)}} \tag{1.6}$$

où $vol(Y)$ désigne le volume d'un domaine Y.

La nouvelle réalisation dépend bien des paramètres régissant la forme et la taille des domaines. En toute généralité, résoudre un problème de calage d'historique avec ce choix de paramétrisation revient à résoudre un problème d'optimisation de forme qui peut s'avérer très complexe. Pour limiter le temps de calcul on choisit des formes simples préétablies qui ne varient que grâce à un nombre restreint de paramètres. Typiquement, on peut choisir pour domaines des cercles de centre fixé (en général au niveau des puits). Dans ce cas, chaque zone est contrôlée par un unique paramètre : son rayon.

Il est ici aussi possible d'étendre cette paramétrisation au cas où l'on veut combiner $N+1$ bruits blancs Gaussiens dans M domaines paramétrés. Pour tout $i \in \{1, \ldots, M\}$ et pour toutes les mailles de modèle X, on appelle $V_i(X)$ le volume de X à l'intérieur du domaine R_i et $V_0 = vol(X - \cup_i R_i)$. Si $t = (t_1, ..., t_q)$ est l'ensemble des paramètres déterminant la forme des domaines, on peut combiner les bruits blancs Gaussiens comme suit :

$$Z(X,t) = \sum_{i=0}^{M} a_i(X,t) Z_{J(i)}(X) \tag{1.7}$$

où $a_i(X,t) = \frac{V_i(X,t)}{\sqrt{\sum_{i=1}^{M} V_i^2(X,t)}}$, $J(i) \in [0, \ldots, N]$ est l'indice du bruit blanc Gaussien associé à la zone R_i et $J(0) = 0$. Pour toutes les mailles X, les volumes $V_i(X,t)$ $(i = 1, ..., M)$ sont bien dépendants du jeu de paramètres t. Par conséquent, le nouveau bruit blanc Gaussien Z est bien paramétré par t. Il est intéressant de noter que cette méthode ne contraint pas la taille des zones. Il est donc possible que les zones se chevauchent. Dans ce cas une combinaison des réalisations correspondant aux zones concernées est choisie, comme illustrée en Figure 1.5.

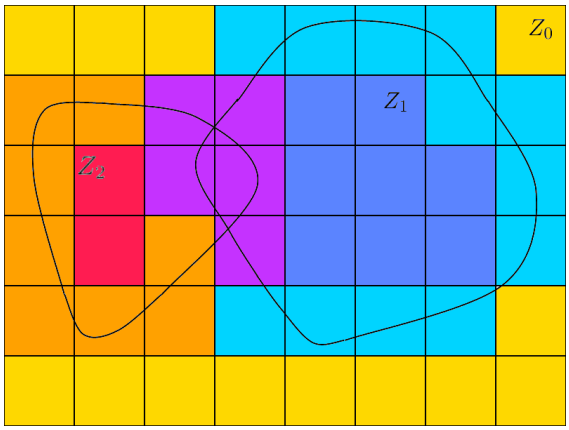


FIGURE 1.5 – Illustration de la méthode de déformation de domaines lorsque plusieurs domaines sont définis.

L'avantage de cette technique de paramétrisation est le fait qu'elle n'est que peu dépendante de la sélection initiale des zones. En effet, pour la méthode de déformation graduelle locale, une mauvaise définition des zones peut avoir un impact très négatif sur la décroissance potentielle de la fonction objectif. La Figure 1.6 issue de [37] illustre bien ce phénomène. Elle compare les résultats de calages d'historiques réalisés avec les méthodes de déformation graduelle locale et de déformation de domaines. Sur cette figure, la courbe rose présente les variations de la fonction objectif avec la taille des régions lorsqu'une réalisation est attribuée à chaque zone. La courbe bleue, quant à elle, présente à chaque point le résultat d'une optimisation de la fonction objectif avec une méthode de déformation graduelle locale pour des régions de taille fixée. Pour chaque taille de région, la méthode de déformation graduelle locale permet bien une certaine diminution de la fonction objectif mais elle ne permet de descendre sous l'optimum de la méthode de déformation de domaines (barre rose) que si les zones sont très bien choisies. La méthode de déformation de domaines est donc meilleure si les zones ne sont pas parfaitement déterminées, ce qui arrive fréquemment dans la pratique.

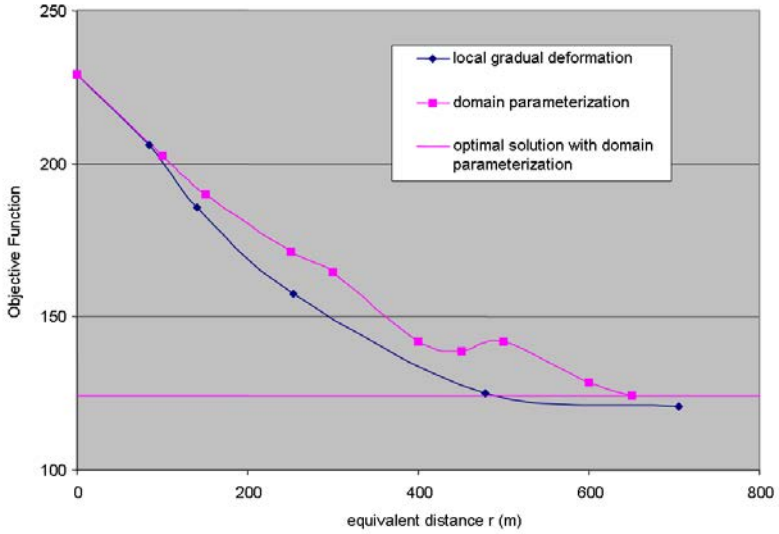


FIGURE 1.6 – Comparaison entre la méthode de déformation de domaines et la méthode de déformation graduelle locale suivant la taille des régions [37].

1.2 Généralisation de la méthode de déformation graduelle

1.2.1 Une autre façon d'écrire la déformation de domaines

Afin de pouvoir combiner les méthodes de déformation graduelle locale et de déformation de domaines, l'écriture de la méthode de déformation de domaines a été modifiée. On commence par un cas simple avec uniquement une zone et deux bruits blancs Gaussiens à combiner.

(i) Une zone et deux bruits blancs Gaussiens standards
On considère une zone R_1 paramétrée par le jeu de paramètres $t = (t_1, ..., t_q)$ et deux réalisations de modèles indépendantes associées aux deux bruits blancs Gaussiens Z_0 et Z_1. Comme auparavant, pour une maille de modèle X donnée, on définit $V_1(X,t)$ comme étant le volume de la maille X à l'intérieur de R_1 et $V_0(X,t) = vol(X - R_1)$ son complémentaire. On définit aussi α_1 par :

$$\alpha_1(X,t) = \frac{V_1(X,t)}{V_1(X,t) + V_0(X,t)} \tag{1.8}$$

Un nouveau bruit blanc Gaussien Z peut alors être construit par :

$$Z(X) = \cos(\frac{\pi}{2}\alpha_1)Z_0(X) + \sin(\frac{\pi}{2}\alpha_1)Z_1(X) \tag{1.9}$$

Ce bruit blanc Gaussien associe Z_1 aux mailles à l'intérieur de R_1 et Z_0 à celles à l'extérieur. Les mailles partiellement à l'intérieur de R_1 se voient quant à elles attribuer une combinaison linéaire des deux bruits blancs Gaussiens. La nouvelle réalisation de modèle $G(Z)$ est bien dépendante de la forme et de la taille de R_1.

(ii) Généralisation : M zones et $M+1$ bruits blancs Gaussiens
On peut généraliser la formulation précédente au cas où M zones sont définies. Si on dispose de $M+1$ bruits blancs Gaussiens indépendants $(Z_0, ..., Z_M)$ et d'un jeu de paramètres t qui paramétrise les formes et tailles de toutes les zones, on peut associer une réalisation de modèle à chaque zone et une autre à l'extérieur de toutes les zones. De façon similaire à l'exemple précédent, on définit $V_i(X,t) = vol(X \cap R_i)$, $V_0(X,t) = vol(X - \cup_j R_j)$ et α_i :

$$\alpha_i(X,t) = \frac{V_i(X,t)}{\sum_{j=0}^{M} V_j(X,t)} \tag{1.10}$$

Pour simplifier les notations, pour tout $i \in \{1, ..., M\}$ on choisit d'associer le bruit blanc Gaussien Z_i à la zone R_i et le bruit blanc Gaussien Z_0 à l'extérieur de toutes les zones. Le nouveau bruit blanc Gaussien Z peut ainsi être construit grâce à l'expression :

$$Z(X) = \prod_{j=1}^{M} \cos(\frac{\pi}{2}\alpha_j)Z_0(X) + \sum_{j=1}^{M} \sin(\frac{\pi}{2}\alpha_j) \left(\prod_{k=j+1}^{M} \cos(\frac{\pi}{2}\alpha_j) \right) Z_j(X) \tag{1.11}$$

On a donc une nouvelle façon de construire une réalisation de modèle $G(Z)$ dépendant uniquement de la forme des domaines définis. On peut vérifier simplement que, comme dans la formulation précédente, pour une maille de modèle X :

- Si X est entièrement et uniquement à l'intérieur de la zone R_i ($\alpha_i = 1$et $\forall j \neq i$, $\alpha_j = 0$), $Z(X) = Z_i(X)$;
- Si X est entièrement à l'extérieur de toutes les zones, $Z(X) = Z_0(X)$;

— Si X est partiellement à l'intérieur d'une ou plusieurs zones, $Z(X)$ est une combinaison linéaire des bruits blancs associés à ces zones.

Avec cette formulation, les paramètres de déformation de domaines sont plus simples à mettre en rapport avec les paramètres de déformation graduelle.

1.2.2 Combinaison de la méthode de déformation graduelle locale avec la méthode de déformation de domaine

Les méthodes de déformation graduelle locale et de déformation de domaines présentent toutes deux des inconvénients. Il est d'abord difficile de choisir convenablement les zones pour la méthode de déformation graduelle et d'assurer une bonne décroissance potentielle pour la fonction objectif. D'un autre côté, la méthode de déformation de domaines ne permet pas la modification des réalisations à l'intérieur des domaines sans chevauchement. On propose dans cette section une méthode qui généralise la déformation graduelle locale en permettant la modification dynamique de la forme des zones en cours d'optimisation. On présentera en premier lieu deux cas simples avant de donner le cas général.

(i) Un exemple simple : une zone et deux bruits blancs Gaussiens

Supposons qu'on dispose de deux bruits blancs Gaussiens (Z_0, Z_1) et d'une zone R_1 sur un modèle de réservoir G. On propose de construire un nouveau bruit blanc Gaussien Z qui dépend d'un paramètre de déformation graduelle ρ^1 et d'un jeu de paramètres t contrôlant la forme de R_1 :

$$Z(X) = \cos(\pi\alpha_1(X,t)\rho^1)Z_0(X) + \sin(\pi\alpha_1(X,t)\rho^1)Z_1(X) \tag{1.12}$$

où $\alpha_1(X,t)$ est donné par l'équation (1.8).

Selon la position d'une maille X sur le modèle de réservoir et la valeur du paramètre de déformation graduelle, on a :

— Si X est entièrement à l'extérieur de R_1 alors $\alpha_1 = 0$ et $Z(X) = Z_0(X)$.

— Si X est entièrement à l'intérieur de R_1 alors $\alpha_1 = 1$ et

$$Z(X) = \cos(\pi\rho^1)Z_0 + \sin(\pi\rho^1)Z_1 \tag{1.13}$$

Ce qui correspond à la méthode de déformation graduelle.

— Si X est partiellement à l'intérieur de R_1 $(0 < \alpha_1 < 1)$, $Z(X)$ est une combinaison de $Z_0(X)$ et $Z_1(X)$. Dans ce cas, les méthodes de déformation graduelle locale et de déformation de domaines sont appliquées simultanément.

— Si le paramètre de déformation graduelle est fixé à $\rho^1 = \frac{1}{2}$, alors $Z(X) = \cos(\frac{\pi}{2}\alpha_1)Z_0(X) + \sin(\frac{\pi}{2}\alpha_1)Z_1(X)$, Ce qui correspond à la méthode de déformation de domaines décrite en section 1.2.1 (équation(1.9)).

On est bien parvenu sur ce cas simple à construire une réalisation de modèle $G(Z)$ qui dépend à la fois d'un paramètre de déformation graduelle et de paramètres régissant la forme d'un domaine locale.

(ii) Un domaine et $N+1$ bruits blancs Gaussiens

On peut étendre le cas précédant en ajoutant la possibilité de combiner $N+1$ bruits blancs Gaussiens standards indépendants $(Z_0, ..., Z_N)$ à l'intérieur de la zone R_1. Si t est toujours le jeu de paramètres contrôlant la forme de R_1 et qu'on introduit N paramètres de déformation graduelle $(\rho^1, ..., \rho^N)$, on peut construire pour toutes les mailles X du modèle un nouveau bruit

blanc Gaussien :

$$Z(X) = \prod_{j=1}^{N} \cos(\pi\alpha_1\rho^j) Z_0(X) + \sum_{j=1}^{N} \sin(\pi\alpha_1\rho^j) \left(\prod_{k=j+1}^{N} \cos(\pi\alpha_1\rho^k) \right) Z_j(X) \tag{1.14}$$

On a alors, selon la position de X dans le modèle de réservoir :

— Si X est entièrement à l'extérieur de R_1, alors $\alpha_1 = 0$ et $Z(X) = Z_0(X)$.

— Si X est entièrement à l'intérieur de R_1, alors $\alpha_1 = 1$ et

$$Z(X) = \prod_{j=1}^{N} \cos(\pi\rho^j) Z_0(X) + \sum_{j=1}^{N} \sin(\pi\rho^j) \left(\prod_{k=j+1}^{N} \cos(\pi\rho^k) \right) Z_j(X) \tag{1.15}$$

ce qui correspond à la formule de la méthode de déformation graduelle décrite dans l'équation (1.2).

— Si X est partiellement à l'intérieur de R_1, $Z(X)$ est une combinaison de $Z_0(X)$ et $Z_1(X)$ crée par l'application simultanée des méthodes de déformation graduelle locale et de déformation de domaine.

Comme dans le cas précédent, on a bien réussi à construire une réalisation de modèle $G(Z)$ qui dépend à la fois d'un paramètre de déformation graduelle et de paramètres régissant la forme d'un domaine locale.

(iii) Cas général : $N+1$ réalisations de modèle et M domaines

On peut maintenant étendre notre méthode au cas général dans lequel on dispose de $N+1$ bruits blancs Gaussiens indépendants $(Z_0, ..., Z_N)$ à combiner et de M domaines $(R_1, ..., R_M)$ définis sur un modèle de réservoir. Pour chaque zone R_i, on définit α_i par l'équation (1.10). Pour chaque couple "zone"/"bruit blanc Gaussien" R_i/Z_j, on introduit un paramètre de déformation graduelle ρ_i^j. On propose la formulation généralisée

$$Z(X) = \prod_{j=1}^{N} \cos(\pi \sum_{i=1}^{M} \alpha_i\rho_i^j) Z_0(X) + \sum_{j=1}^{N} \sin(\pi \sum_{i=1}^{M} \alpha_i\rho_i^j) \prod_{k=j+1}^{N} \cos(\pi \sum_{i=1}^{M} \alpha_i\rho_i^k) Z_j(X) \tag{1.16}$$

Cette formule permet d'effectuer une déformation graduelle de toutes les réalisations de modèle à l'intérieur de toutes les zones. Généralement, le nombre de paramètres introduits $(M*(N+1))$ est bien trop grand pour être utilisable dans la pratique. Cependant, il est possible de réduire le nombre de paramètres simplement en sélectionnant les "meilleures" réalisations de modèle à combiner dans chaque zone. Cette sélection peut se faire en fixant certains des paramètres ρ_i^j à 0.

Analysons les cas possibles pour une maille de modèle donnée X :

1. *A l'extérieur de tous les domaines*
   Pour chaque i, $\alpha_i = 0$. Donc, $Z(X) = Z_0(X)$.
2. *Suppression de l'influence de la réalisation l dans la zone R_q*
   Fixons le paramètre ρ_q^l à 0. Si la maille X est entièrement et uniquement dans le domaine R_q, $Z(X)$ devient :

$$Z(X) = \prod_{j=1, j\neq l}^{N} \cos(\pi\rho_q^j) Z_0(X) + \sum_{j=1, j\neq l}^{N} \sin(\pi\rho_q^j) \prod_{k=j+1}^{N} \cos(\pi\rho_q^j) Z_j(X) \tag{1.17}$$

   La réalisation l n'a plus d'influence dans la zone R_q.

3. *Déformation de domaines de la zone R_q avec seulement la réalisation de modèle l associée à R_q*

   On souhaite appliquer Z_l à l'intérieur de R_q. Fixons $\rho_q^l = \frac{1}{2}$ et pour tous $j \in \{1, \ldots, l-1, l+1, \ldots, N\}$ les paramètres $\rho_q^j = 0$. Alors :
   — Si X est entièrement et uniquement à l'intérieur de R_q alors $Z(X) = Z_l(X)$;
   — Si X est seulement partiellement à l'intérieur de R_q mais à l'extérieur des autres zones, alors une combinaison de Z_0 et Z_l est appliquée, de la même façon que pour la méthode de déformation de domaines (équation (1.11)).

   Ce cas correspond à une déformation de domaines appliquée à la zone R_q.

4. *Déformation graduelle de la zone R_q*

   Si aucun paramètre contrôlant la forme de R_q n'est introduit, alors pour toutes les mailles de modèle X, $\alpha_q(X)$ est constant. Cela revient à effectuer une déformation graduelle locale à l'intérieur de la zone R_q.

Nous sommes bien parvenus à généraliser la méthode de déformation graduelle locale de façon à permettre de déformer les zones dynamiquement durant le processus de calage d'historique. Cette nouvelle méthode offre même une très grande flexibilité concernant la définition des différents domaines et le choix des réalisations de modèle à combiner dans chacun d'eux. En effet, toute combinaison des méthodes de déformation graduelle et de déformation de domaines est possible. Par exemple, combiner deux ou plus réalisations à l'intérieur d'une zone de forme fixe tout en ne déformant que la forme d'une autre est tout à fait concevable.

Les tests numériques réalisés dans la suite du document exploitent la possibilité de n'utiliser qu'un nombre restreint de paramètres de déformation graduelle. Si on dispose de $N+1$ bruits blancs Gaussiens à combiner et de M domaines sur un modèle de réservoir, on procède de la manière suivante :
— choisir un bruit blanc Gaussien de base Z_0 qu'on applique en dehors de toutes les zones ;
— choisir une réalisation de modèle à appliquer pour chaque zone ;
— combiner dans chaque zone la réalisation de base avec la réalisation associée.

En clair, si le bruit blanc Gaussien Z_j est associé à la zone R_i, ρ_i^j est choisi comme paramètre du problème tandis que tous les autres ρ_i^k tels que $k \neq j$ sont fixés à 0. Si un paramètre est introduit pour contrôler la forme de chaque zone, on aura donc paramétrisé notre problème de calage d'historique avec $2M$ paramètres.

1.3 Initialisation du problème de calage d'historique : une méthode de patchwork

L'un des problèmes du calage d'historique est son initialisation. Pour les méthodes de paramétrisation géostatistique locales décrites dans la section précédente, il est crucial de choisir des domaines pertinents pour avoir une bonne décroissance de la fonction objectif lors du processus d'optimisation. De la même façon, le choix des réalisations de modèle à combiner pour la méthode de déformation graduelle conditionne de façon importante la décroissance potentielle de la fonction objectif. Enfin, choisir un jeu de paramètres initial proche du point optimal peut s'avérer important pour assurer le succès et limiter le coût de l'algorithme d'optimisation. Il est donc nécessaire de disposer d'une méthode efficace d'initialisation des paramètres du problème.

Après avoir présenté une méthode d'initialisation du problème par patchwork, nous proposons dans cette section une nouvelle méthodologie.

1.3.1 Une méthode de patchwork classique pour initialiser le problème de calage d'historique

Lorsqu'il est possible de séparer un modèle de réservoir en plusieurs zones relativement indépendantes, il est aussi possible d'initialiser le problème en choisissant dans chacune de ces zones une réalisation différente comme présenté dans [90]. Le choix de la réalisation à appliquer dans chaque zone est en général fait en comparant la qualité du calage des données de production à chaque puits de la zone. La Figure 1.7 montre un exemple de modèle de réservoir séparé en 11 zones censées être relativement indépendantes. Si les zones sont réellement indépendantes (c'est à dire si la modification de la réalisation de modèle à l'intérieur d'une zone n'affecte pas le comportement du réservoir dans une autre), une initialisation par patchwork doit proposer une réalisation de modèle présentant une meilleure valeur de fonction objectif qu'une réalisation aléatoire. Cependant, si les zones ne sont pas réellement indépendantes, il n'est pas possible d'assurer qu'une initialisation par patchwork ne détériorera pas le modèle.

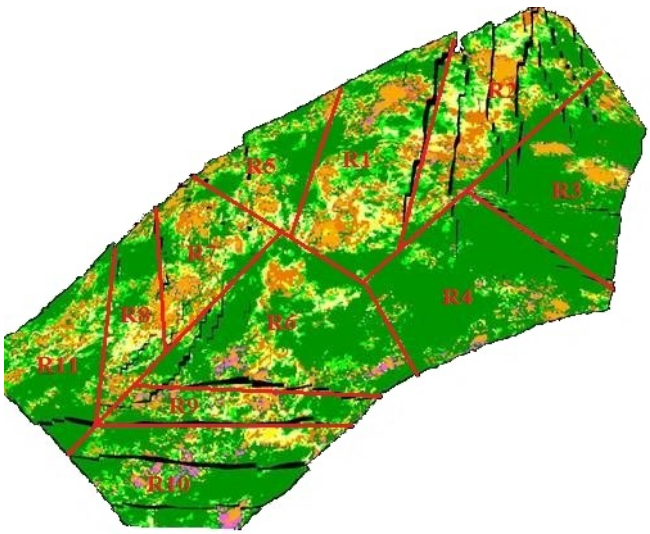


FIGURE 1.7 – Séparation d'un réservoir en 11 zones supposées indépendantes.

Décrivons plus précisément une méthodologie de patchwork. Il est possible d'écrire la fonction

objectif sous la forme :

$$F(x_1, ..., x_p) = \sum_{i=1}^{n} f_i(x_1, \ldots, x_p) \tag{1.18}$$

où pour tout i, f_i est la fonction objectif associée au i^{eme} puits sur lequel il existe des données de production à caler, n est le nombre de puits sur le réservoir et $(x_1, \ldots, x_p)$ sont les paramètres du problème de calage d'historique. Une fois que le modèle de réservoir a été séparé en M zones relativement indépendantes $(R_1, ..., R_M)$, on note I_k l'ensemble des indices des puits appartenant à la zone R_k. On appelle alors fonction objectif locale associée à la zone R_k la fonction :

$$g_k = \sum_{i \in I_k} f_i$$

Une initialisation par patchwork peut alors être effectuée en suivant les étapes suivantes :

1. Générer $N + 1$ réalisations de modèle.
2. Créer M zones indépendantes dans le réservoir.
3. Appeler Z_0 le bruit blanc Gaussien associé à la réalisation de modèle présentant la plus basse valeur de la fonction objectif F.
4. Pour chaque zone R_k :
   — trier les valeurs de la fonction objectif locale f_k. Soit Z_{j_k} le bruit blanc Gaussien tel que $g_k(Z_{j_k})$ est minimal,
   — si $Z_{j_k} \neq Z_0$, appliquer Z_{j_k} dans la zone R_k. Z_{j_k} sera combiné ultérieurement à Z_0 dans la zone R_k par déformation graduelle locale.
5. Appliquer Z_0 dans toutes les mailles de modèle n'ayant pas été déjà associées à une réalisation de modèle.

La Figure 1.8 illustre cette méthode de patchwork. Quatre zones supposées indépendantes ont été créées sur un modèle de réservoir comprenant 12 puits $(P1, ..., P12)$. Quatre réalisations de modèle ont été générées. Après le traitement décrit ci-dessus, Z_0 est choisi comme étant le bruit blanc associé à la meilleure réalisation, la réalisation associée à Z_3 est appliquée dans la zone supérieure gauche, celle associée à Z_1 est appliquée dans la zone supérieure droite tandis que celle associée à Z_2 est appliquée dans la zone inférieure droite. La meilleure réalisation locale pour la zone inférieur droite est Z_0, ses mailles se voient donc attribuer la réalisation de base Z_0. Si l'hypothèse d'indépendance des zones est correcte, la réalisation de modèle ainsi construite donnera une meilleure valeur de la fonction objectif que celle associée à Z_0.

1.3.2 Amélioration de la méthode de patchwork

Avec notre nouvelle méthode de paramétrisation ou la méthode de déformation de domaines, le fait que les formes des domaines puissent varier de façon continue offre une plus grande souplesse pour le processus d'initialisation du problème. On améliore ainsi la méthode de patchwork comme suit :

1. Générer $N + 1$ réalisations de modèle.
2. Appeler Z_0 le bruit blanc Gaussien associé à la réalisation de modèle présentant la plus faible valeur de fonction objectif.
3. Associer les puits en groupes. Pour un groupe de puits I, on appelle $g_I = \sum_{i \in I} f_i$ la fonction objectif locale associée.
4. Pour chaque groupe de puits I :

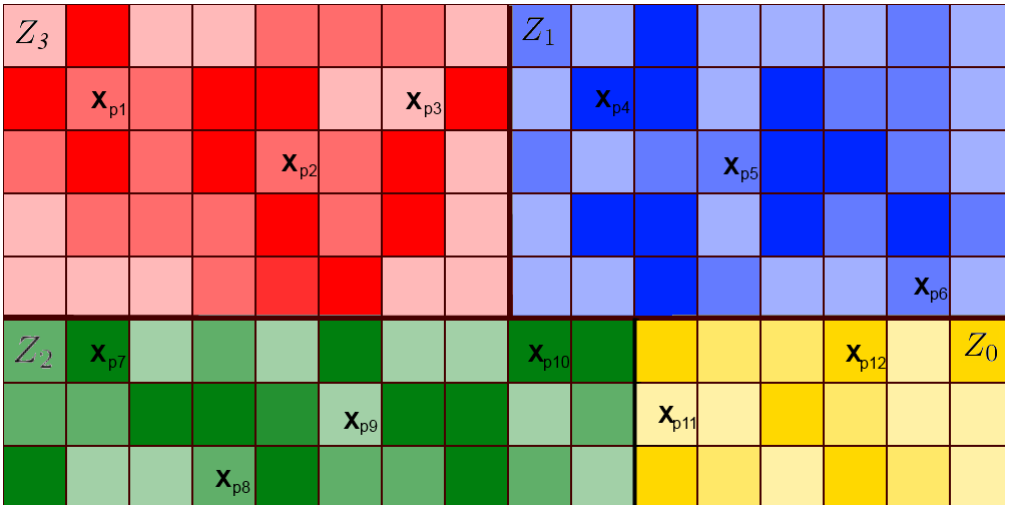


FIGURE 1.8 – Illustration de la méthode de patchwork classique.

- trier les valeurs des fonctions objectifs locales $g_I(Z_j)$. Soit Z_{j_I} le bruit blanc Gaussien tel que $g_I(Z_{j_I})$ soit minimale,
- si $Z_{j_I} \neq Z_0$, définir un zone paramétrée contenant tous les puits de I et appliquer Z_{j_i} à l'intérieur. La zone peut par exemple être circulaire et paramétrée par son seul rayon r.

5. Effectuer une optimisation simple avec un unique paramètre contrôlant la taille de toutes les zones. Dans le cas de zones circulaires, on peut optimiser les tailles des zones avec un unique paramètre de rayon.

Comparée à la méthode de patchwork décrite précédemment et proposée dans [90], les domaines paramétrés définis dans l'étape 4 peuvent prendre une forme quelconque au lieu d'être limités par les mailles du modèle. De plus, l'étape 5 d'optimisation permet d'améliorer la taille des zones pour garantir une diminution de la fonction objectif. On verra dans les exemples numériques suivants que pour un coût très limité (quelques simulations d'écoulement), cette méthode permet d'obtenir une très bonne initialisation du problème.

La Figure 1.9 illustre la nouvelle méthode de patchwork. Sur ce modèle de réservoir, quatre réalisations de modèles ont été générées et chaque groupe de puits contient un et un seul puits. La meilleure réalisation de modèle, associée à Z_0, est appliquée en dehors de toutes les zones. De plus, trois zones ont été créées autour des puits $P2$, $P5$ et $P19$. Ces tois zones se sont vues attribuer respectivement les réalisations de modèle associées aux bruits blancs Gaussiens Z_3, Z_1 et Z_2. Chacune de ces zones est circulaire et peut donc être paramétrée par son rayon. L'étape d'optimisation préliminaire (étape 5) permet d'ajuster les tailles de ces zones dans le cas où la réalisation créée ne donne pas de meilleurs résultats que celle associée à Z_0.

Dans la suite du document, toutes les zones considérées seront circulaires et paramétrées par leur rayon respectif.

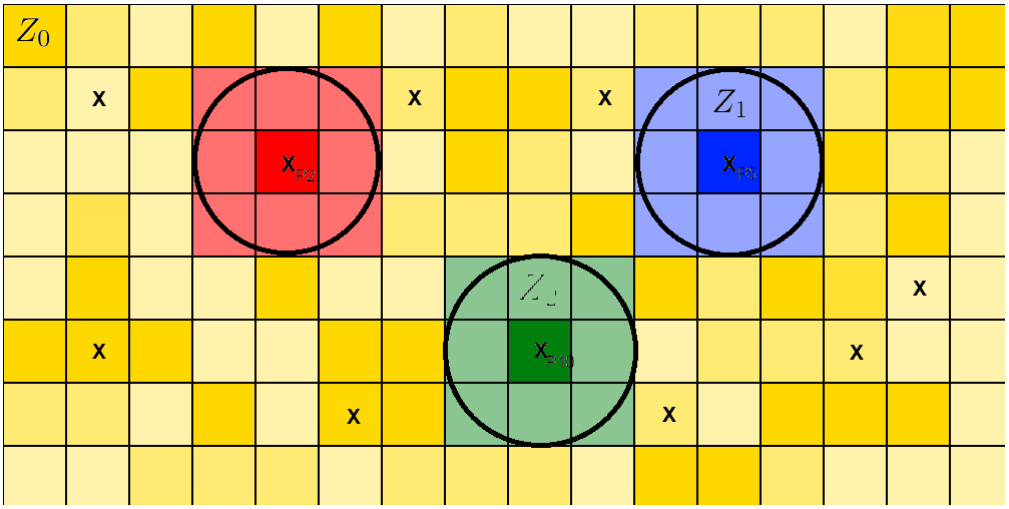


FIGURE 1.9 – Illustration de la méthode de patchwork améliorée.

1.4 Validation numérique des méthodes présentées

Les méthodes de paramétrisation et d'initialisation introduites dans les sections précédentes ont été codées en C et intégrées à l'outil IFP énergies nouvelles CONDOR. On présente ici les résultats numériques réalisés sur un cas synthétique de réservoir simple.

1.4.1 Description du cas test

On décrit dans cette section le cas synthétique de réservoir qui sera utilisé pour les descriptions et tests des méthodes proposées dans la suite.

On considère le champ représenté en Figure 1.10 utilisé auparavant dans [36]. Ses caractéristiques sont les suivantes :

1. 2500 m dans les directions x et y.
2. 10 m dans la direction z.
3. Discrétisé uniformément par 50x50x1 mailles de 50 m dans les directions x et y et de 10 m dans la direction z.
4. 25 puits verticaux :
   — 13 producteurs (P1,P3,...,P25)
   — 12 injecteurs (I2,I4,...,I24)
5. Réservoir hétérogène : longueur de corrélation de 150 m dans les directions x et y
6. Pression initiale dans le réservoir de 250 bar.
7. Pression en fond de puits imposée :
   — 320 bar pour les injecteurs
   — 180 bar pour les producteurs

Un historique de données de production de 4000 jours est connu.

1.4.2 Initialisation

Cinq jeux de 25 réalisations aléatoires ont été générés pour appliquer la méthode de patchwork améliorée avec des domaines radiaux autour des puits. On donne les valeurs des fonctions objectifs et fonctions objectifs locales pour la réalisation de base Z_0 (la meilleure réalisation

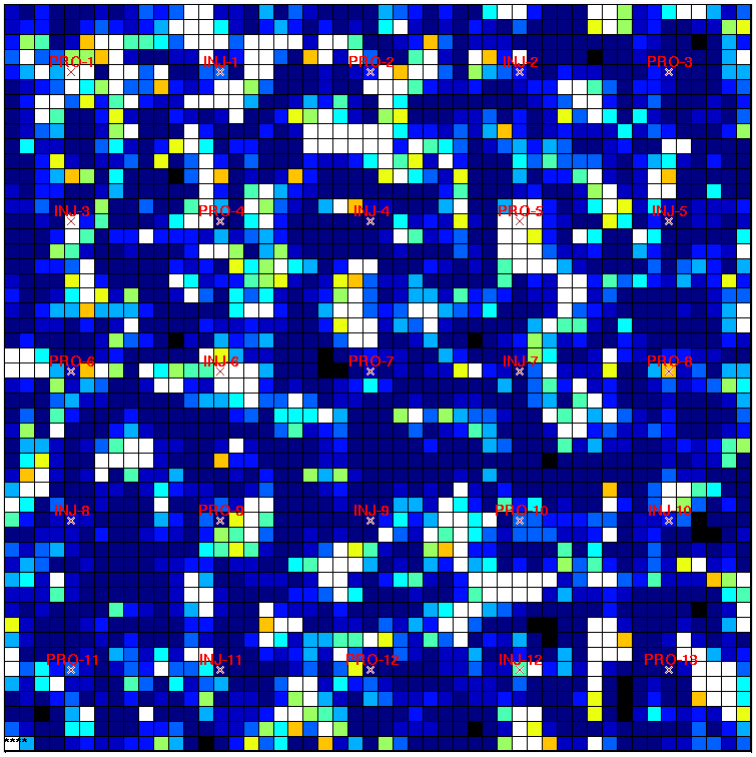


FIGURE 1.10 – Carte de perméabilité du champ synthétique comprenant 13 puits producteurs et 12 puits injecteurs [37].

parmis les 25 générées) et quatre autres réalisations aléatoires pour un des cas testés dans le Tableau 1.1. On peut facilement voir qu'il y a de grandes disparités entre les valeurs des fonctions objectif locales des différentes réalisations. Z_0 est en effet bien meilleure que les autres globalement (fonction objectif globale de 2543.72 contre plus de 7000 pour toutes les autres), mais certaines des autres réalisations peuvent être meilleures localement. Un bon exemple est de comparer les valeurs des fonctions objectif locales du puits producteur P-12 (7ème colonne, 4ème ligne de chaque réalisation dans le tableau). Pour ce puits, la fonction objectif locale pour la réalisation associée à Z_0 est de 175.33 alors qu'elle ne vaut 15.21 pour la réalisation associée à Z_2, soit un gain relatif de 91% sur ce puits. La méthodologie de patchwork proposée va tendre à créer des zones autour des puits correspondant à ce cas de figure.

Réalisation	F						
Z_0	2543.72	26.37	7.00	110.55	31.13	80.98	52.10
		167.24	11.45	3.83	4.50	67.02	155.68
		153.42	82.50	8.11	28.15	114.06	44.54
		334.34	38.25	170.00	38.75	<u>175.33</u>	128.42
		509.87					
Z_1	16747.78	3114.85	325.78	34.00	30.18	157.73	2121.088
		19.23	50.72	146.70	35.57	11.07	1370.06
		108.09	282.76	140.77	10.21	4309.95	12.86
		687.70	55.46	10.24	132.39	<u>48.48</u>	3053.02
		478.77					

Z_2	47728.54	398.84	3.94	37.45	14.37	142.63	19.60
		46.04	29.01	134.79	14.03	54.10	1.44
		363.70	374.83	86.37	401.38	32553.64	23.39
		67.40	43.03	2000.42	1159.04	15.21	9231.61
		512.17					
Z_3	10012.71	430.83	3.43	92.43	26.53	75.05	278.32
		8.82	1.86	47.31	0.96	27.56	17.75
		80.97	318.86	423.99	82.64	557.05	39.49
		68.49	58.50	8.02	674.52	40.07	6246.31
		402.84					
Z_4	7058.64	22.33	43.51	26.58	3.62	33.63	20.42
		379.42	27.68	150.12	14.54	138.80	15.84
		844.50	352.01	119.07	309.90	2477.19	47.45
		218.51	50.53	31.10	266.73	90.09	873.89
		501.08					

TABLE 1.1 – Valeurs de la fonction objectif et des fonctions objectif locales pour quatre réalisations de modèle.

D'un autre côté, on peut remarquer que certaines valeurs de fonctions objectif locales sont très petites devant la valeur de la fonction objectif globale pour la réalisation associée à Z_0 (pour le puits I-1, la valeur de la fonction objectif locale vaut moins de 1% de la fonction objectif globale). Afin de limiter le nombre de paramètres introduits, on essaie de limiter la création de zones autour des puits ne présentant qu'une très faible fonction objectif locale pour Z_0 puisque le gain potentiel devient trop faible par rapport au coût d'optimisation de ces paramètres. On a donc pour cet exemple modifié légèrement la méthodologie présentée en section 1.3.2. La nouvelle méthodologie devient :

1. Générer $N + 1$ réalisations de modèle.
2. Appeler Z_0 le bruit blanc Gaussien associé à la réalisation de modèle présentant la plus faible valeur de fonction objectif.
3. Pour chaque puits i :
   — trier les valeurs des fonctions objectifs locales $f_i(Z_j)$. Soit Z_{j_i} le bruit blanc Gaussien tel que $f_i(Z_{j_i})$ soit minimale,
   — si $f_i(Z_0) > \alpha F(Z_0)$ et $f_i(Z_{j_i}) < f_i(Z_0)$, avec $0 < \alpha < 1$ une constante fixée au préalable, définir un zone paramétrée centrée autour du puits i et appliquer Z_{j_i} à l'intérieur. La zone peut par exemple être circulaire et paramétrée par son seul rayon r.
4. Effectuer une optimisation simple avec un unique paramètre contrôlant la taille de toutes les zones.

Dans ce cas, on a choisi α à 1%.

La Figure 1.11 illustre la nouvelle méthode de patchwork en montrant les zones circulaires créées pour l'un des 4 cas étudiés. Sur cette figure, les cercles représentent les zones créées. Ici, la méthodologie décrite précédemment prescrit la création de 16 zones. Le problème de calage d'historique a donc été initialisé avec 32 paramètres :

— 16 paramètres de déformation graduelle
— 16 paramètres de rayon

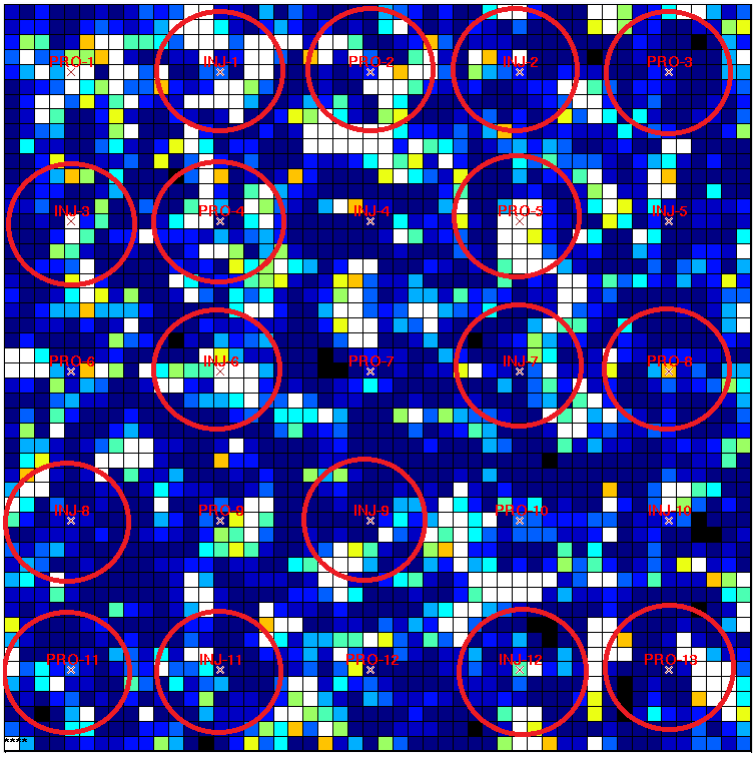


FIGURE 1.11 – Initialisation par méthode de patchwork améliorée.

La méthode de patchwork améliorée a été appliquée à chacun des 5 jeux de 25 réalisations. La Figure 1.12 montre pour chaque cas la valeur de la fonction objectif pour la réalisation de base (la meilleure parmis 25) et la réalisation construite par patchwork avec un rayon de zones fixé à $r = 300m$ (6 mailles de modèles). Sur cette figure, l'étape d'optimisation préliminaire (étape 4 de la méthodologie) n'a donc pas encore été suivie. Pour les quatre premiers cas, une méthode de patchwork apporte déjà un gain significatif à la fonction objectif sans simulation d'écoulements supplémentaire sur le modèle de réservoir : les réalisations de modèle construites présentent respectivement un gain relatif de 83%, 74%, 95% et 78% sur la fonction objectif. Cependant, pour le dernier cas, la réalisation construite par patchwork donne une moins bonne valeur de fonction objectif.
En conséquence, on montre en Figure 1.13 la valeur de la fonction objectif après patchwork en fonction des rayons des zones pour le cas 5. On y observe bien que la fonction objectif a une valeur supérieur à la réalisation de modèle de base (ligne bleue) lorsque $r = 300m$. Toutefois, pour tout $r < 440m$, la méthode de patchwork améliore la réalisation de modèle. La courbe est même convexe et donc très facile à optimiser. Un algorithme de descente fondé sur des approximations du gradient trouve le minimum de cette fonction à $r = 206m$ en seulement 5 simulations d'écoulement dans le réservoir. La fonction objectif passe alors de 10912 pour la réalisation de base à 3987 pour la réalisation optimisée, soit un gain relatif de plus de 63%.

1.4.3 Paramétrisation : Déformation graduelle locale généralisée

Dans chacun des 5 cas précédents, on effectue un calage d'historique des données de production sur le réservoir. Les problèmes ont été initialisés avec les réalisations de modèle construites en section 1.4.2. On compare ici trois méthodes de paramétrisation : la méthode de déformation graduelle locale, la méthode de déformation de domaines et la méthode de déformation graduelle

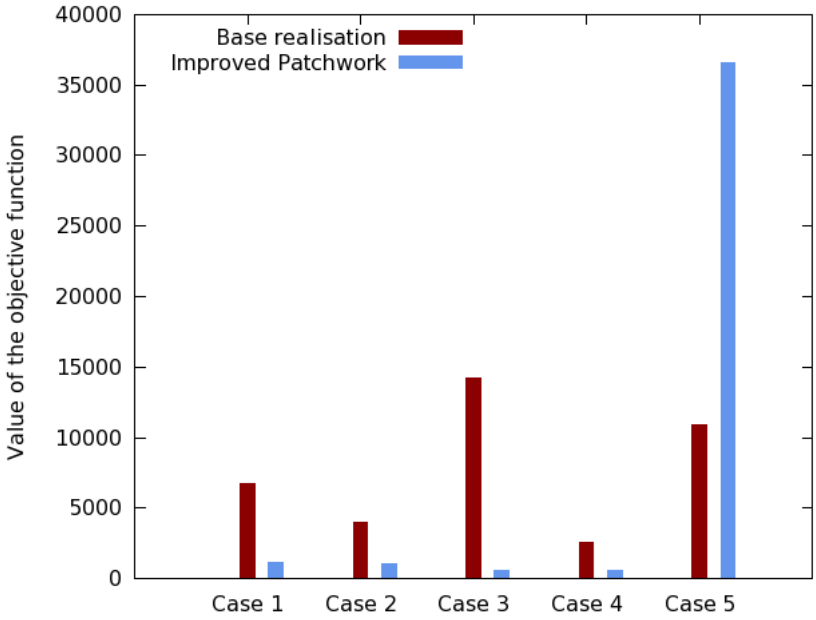


FIGURE 1.12 – Valeur de la fonction objectif pour la meilleure réalisation (en bleu) et pour la réalisation construite via patchwork avec des rayons de $r = 300m$ (en rouge) pour les 5 cas considérés.

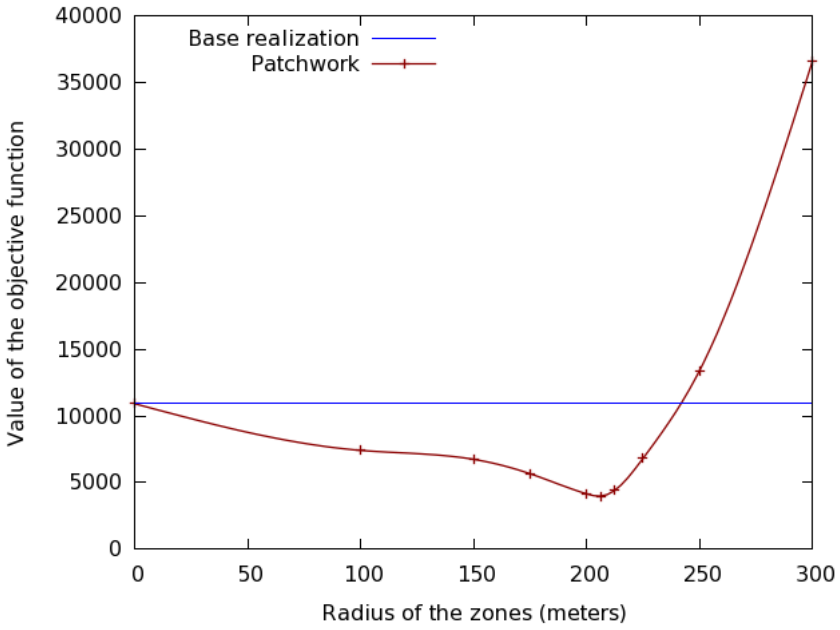


FIGURE 1.13 – Valeur de la fonction objectif en fonction du rayon des zones pour le cas de figure 5.

généralisée.
(i) Déformation graduelle locale : pour chacune des zones créées, on introduit un paramètre de déformation graduelle. On combine alors dans une zone i la réalisation de modèle de base avec la réalisation de modèle qui a été associée à cette zone.

(ii) Déformation de domaines : chaque zone créée conserve son association à une réaliastion de modèle spécifique. On introduit un paramètre de rayon pour chacune.

(iii) Déformation graduelle généralisée : on introduit pour cette méthode deux paramètres pour chaque zone : un paramètre de déformation graduelle et un paramètre contrôlant le rayon de la zone.

Comparaison de la méthode de déformation graduelle locale avec la méthode de déformation graduelle généralisée

On présente en Figure 1.14 les résultats des calages d'historique paramétrisés par méthode de déformation graduelle locale et par méthode de déformation graduelle généralisée. Les optimisations ont été réalisées à l'aide de l'algorithme SQPAL, basé sur des modèles quadratiques locaux construits à partir d'approximations du gradient et de la Hessienne de la fonction objectif. Dans chacun des cas, la vitesse de décroissance de la fonction objectif est similaire pour les deux méthodes. La décroissance est plus rapide par méthode de déformation graduelle généralisée pour les cas 1, 2 et 5, tandis qu'elle est identique ou légèrement plus lente pour les cas 3 et 4.

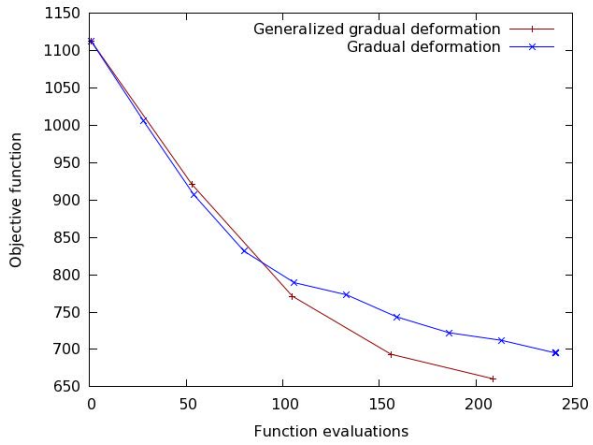
- Cas 1. La déformation graduelle obtient une valeur minimale de fonction objectif en 240 évaluations, la déformation graduelle généralisée parvient à la même valeur en 155 itérations.
- Cas 2. La déformation graduelle obtient une valeur minimale de fonction objectif en 197 évaluations, la déformation graduelle généralisée parvient à la même valeur en 100 itérations.
- Cas 3. La déformation graduelle obtient une valeur minimale de fonction objectif en 150 évaluations, la déformation graduelle généralisée parvient à la même valeur en 150 itérations.
- Cas 4. La déformation graduelle obtient une valeur minimale de fonction objectif en 261 évaluations, la déformation graduelle généralisée parvient à la même valeur en 254 itérations.
- Cas 5. La déformation graduelle obtient une valeur minimale de fonction objectif en 420 évaluations, la déformation graduelle généralisée parvient à la même valeur en 80 itérations.

Malgré le nombre plus élevé de paramètres, il est clair que la nouvelle méthode permet une vitesse de décroissance de la fonction objectif au moins équivalente à la déformation graduelle locale. Dans certains cas, cette vitesse est même bien meilleure.

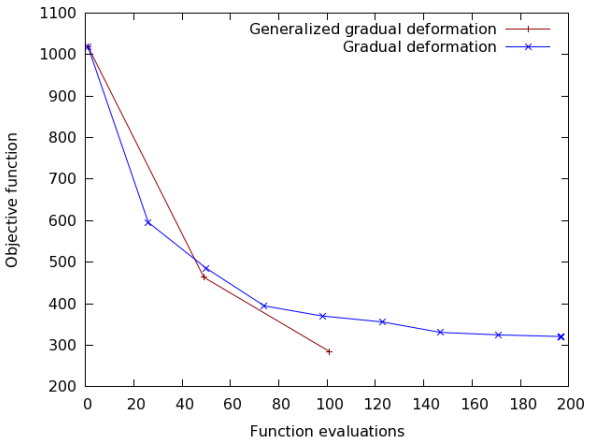
De la même façon, on peut observer que la valeur finale de la fonction objectif est dans tous les cas meilleure avec la nouvelle méthode de paramétrisation qu'avec la méthode de déformation graduelle locale classique. Les gains relatifs sont en effet respectivement pour tous les cas de 5%, 11% , 20%, 47% et 27%. La prise en compte de la taille des zones permet donc une augmentation de la vitesse de décroissance de la fonction objectif ainsi que l'obtention d'une meilleure réalisation de modèle finale.

Comparaison de la méthode de déformation de domaines avec la méthode de déformation graduelle généralisée

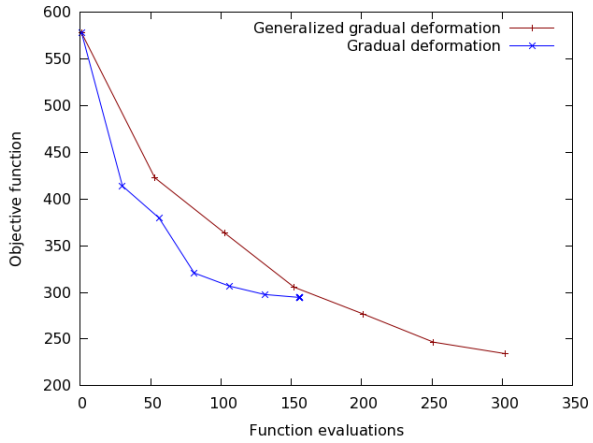
La Figure 1.15 compare les résultats de calages d'historiques réalisés avec les méthodes de déformation graduelle généralisée et de déformation de domaines. Ces résultats sont très similaires à ceux de la Figure 1.14. Ici encore, la méthode de déformation graduelle généralisée présente une vitesse de décroissance de la fonction objectif identique ou légèrement meilleure que la méthode de déformation de domaines tout en donnant une bien meilleure réalisation finale :



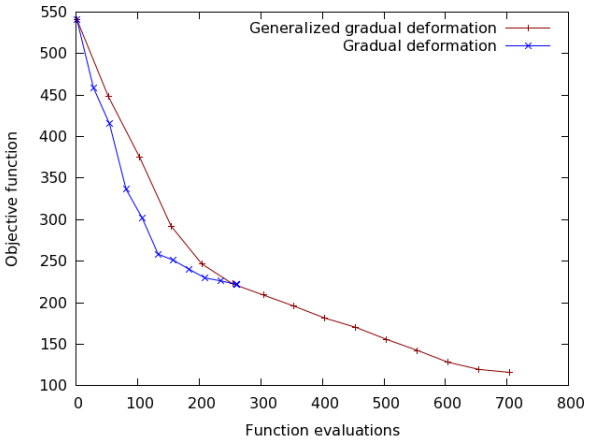
(a) Cas 1



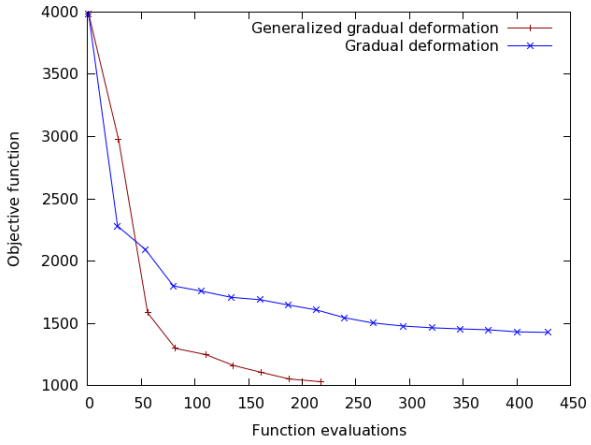
(b) Cas 2



(c) Cas 3



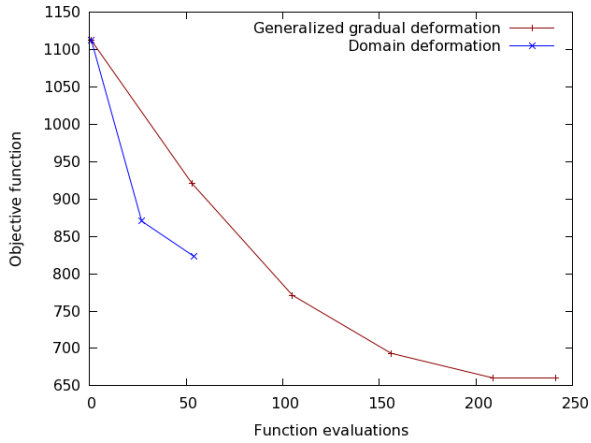
(d) Cas 4



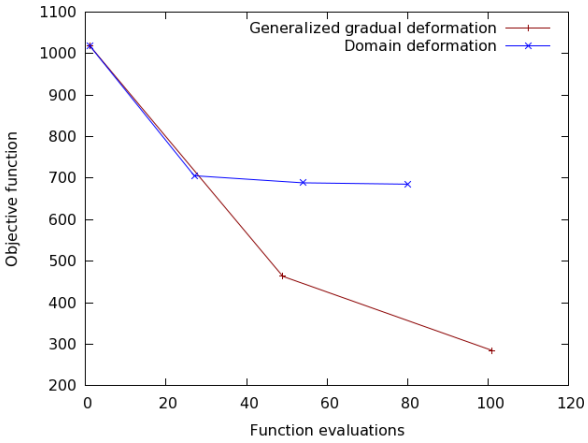
(e) Cas 5

FIGURE 1.14 – Comparaison des méthodes de déformation graduelle locale et de déformation graduelle généralisée sur les 5 cas présentés.

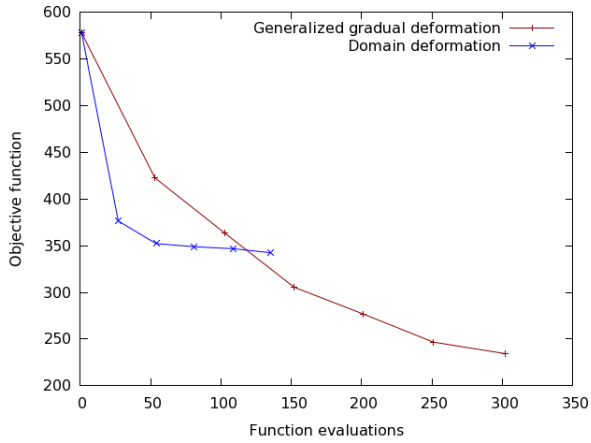
— Cas 1. La déformation de domaines obtient une valeur minimale de fonction objectif en 54 évaluations, la déformation graduelle généralisée parvient à la même valeur entre 50



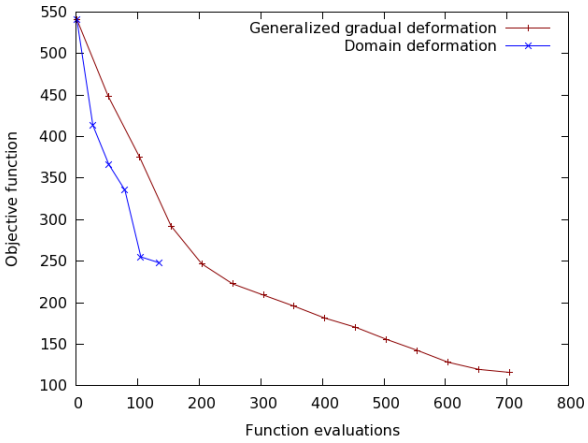
(a) Cas 1



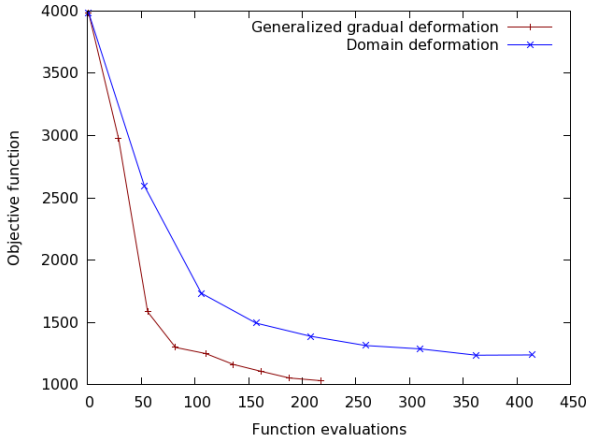
(b) Cas 2



(c) Cas 3



(d) Cas 4



(e) Cas 5

FIGURE 1.15 – Comparaison des méthodes de déformation de domaines et de déformation graduelle généralisée sur les 5 cas présentés.

et 100 évaluations.

— Cas 2. La déformation de domaines obtient une valeur minimale de fonction objectif en 80 évaluations, la déformation graduelle généralisée parvient à une meilleure valeur en 49 évaluations.
— Cas 3. La déformation de domaines obtient une valeur minimale de fonction objectif en 145 évaluations, la déformation graduelle généralisée parvient une meilleure valeur en 152 évaluations.
— Cas 4. La déformation de domaines obtient une valeur minimale de fonction objectif en 134 évaluations, la déformation graduelle généralisée parvient à la même valeur en 204 évaluations.
— Cas 5. La déformation de domaines obtient une valeur minimale de fonction objectif en 414 évaluations, la déformation graduelle généralisée parvient à la même valeur en 110 évaluations.

De la même façon que pour les résultats précédents, l'ajout de paramètres dans la nouvelle méthode de paramétrisation n'a pas affecté la vitesse de décroissance de la fonction objectif comparé à la méthode de déformation de domaines.

La valeur finale de la fonction objectif est quant à elle meilleure dans tous les cas lorsque la paramétrisation a été faite par déformation graduelle généralisée. Les gains relatifs sont en effet de 19%, 57%, 24%, 53% et 16%. Le fait de pouvoir modifier les réalisations à l'intérieur des zones est donc nécessaire dans ce cas à une bonne décroissance de la fonction objectif.

1.5 Bilan du chapitre

Ce chapitre a montré l'importance de disposer de paramètres permettant de modifier les réalisations de modèle de façon pertinente pour le problème de calage d'historique. En effet, les méthodes de paramétrisation géostatistiques de déformation graduelle locale et de déformation de domaines sont capables de donner de bons résultats mais présentent toutes deux des faiblesses qui peuvent limiter la diminution de la fonction objectif. Les performances de la méthode de déformation graduelle locale sont contraintes par les définitions initiales des zones dans lesquelles les réalisations de modèles seront combinées. Celles de la méthode de déformation de domaines sont au contraire limitées par le fait que la réalisation de modèle construite ne peut pas varier à l'intérieur des zones. On a proposé dans ce chapitre une nouvelle méthode de paramétrisation permettant d'allier les qualités des deux méthodes citées ainsi qu'une méthode efficace pour initialiser les paramètres introduits. À noter que des résultats numériques d'application de cette nouvelle méthode sur des cas de réservoir plus complexes seront présentés dans le chapitre 3.

Chapitre 2

Optimisation sans dérivées pour le problème de calage d'historique

Un nouvel algorithme d'optimisation sans dérivées adapté aux fonctions partiellement séparables (DFO-PSOF pour "Derivative Free Optimization for Partially Separable Objective Functions") est présenté dans ce chapitre. Il s'agit d'un algorithme à région de confiance utilisant des modèles d'interpolation quadratique de la fonction objectif. On rappelle qu'une fonction partiellement séparable est une fonction qui peut s'écrire sous la forme :

$$F(x_1, \ldots, x_p) = \sum_{i=1}^{n} f_i(x_{i,1}, \ldots, x_{i,p_i}) \tag{2.1}$$

avec pour tout $i \in \{1, ..., n\}$, $p_i < p$. L'adaptation à la séparabilité partielle de la fonction objectif se fait au niveau de la construction des modèles, similairement à l'adaptation aux fonctions de type moindres carrés [113, 112] : au lieu de construire un modèle pour la fonction objectif globale, la méthode DFO-PSOF construit un modèle pour chaque sous-fonction objectif. Chacune de ces fonctions ne dépendant que d'un nombre très restreint de paramètres, les modèles d'interpolation peuvent en particulier être construits avec seulement un petit nombre de points sans détériorer leur précision.

On présente en section 2.1 un état de l'art des méthodes d'optimisation locales utilisées en ingénierie pétrolière. En section 2.2 sont rappelés les résultats nécessaires à l'introduction du nouvel algorithme. Ce dernier est ensuite décrit en section 2.3 avec la preuve de sa convergence en section 2.4. Enfin, on montre en section 2.5 les premiers résultats numériques obtenus à partir de son implémentation en C++.

2.1 État de l'art

De nombreux tests ont été réalisés dans la littérature et à IFPEN pour déterminer des méthodes d'optimisation efficaces dans le domaine pétrolier afin d'apporter des solutions "optimales" à des problèmes variés tels que le placement des puits [5, 7, 13], la maximisation de la production [73, 31, 114, 109] ou le calage d'historique [11, 18, 36]. Pour simplifier l'écriture, on considère que ces problèmes s'écrivent sous la forme :

$$\begin{gathered} \text{Trouver } (x_1^*, ..., x_p^*) \in [b1, B1] \times ... \times [b_p, B_p] \text{ tel que} \\ (x_1^*, ..., x_p^*) = argmin F(x_1, \ldots, x_p) \end{gathered} \tag{2.2}$$

où $F(x_1, \ldots, x_p) = \sum_{i=1}^{n} f_i(x_1, \ldots, x_p)$ est supposée continûment différentiable avec ∇F continu et Lipschitzien sur le domaine vérifiant les contraintes.

Par exemple, le calage d'historique est un problème de minimisation de l'erreur de production sur chaque puits et peut donc s'écrire sous la forme d'une minimisation d'une somme de fonctions. Les contraintes de bornes sont présentes pour assurer que chacun des paramètres ne varie pas au-delà de tout sens physique ou pour éviter d'ajouter des minimums locaux, la fonctions objectif étant périodique en certains paramètres (les paramètres de déformation graduelle par exemple). La fonction objectif étant par exemple périodique pour les paramètres de déformation graduelle, il est naturel de contraindre ces derniers à rester dans la limite d'une période.

Il existe une très grande diversité de méthodes d'optimisation dans la littérature dont bon nombre ont été appliquées à des problèmes pétroliers. On peut trouver par exemple des méthodes stochastiques [5, 43] ou déterministes [73, 36], des méthodes globales [13, 48, 79] ou locales [11, 70], etc. Les méthodes globales, bien que permettant généralement un meilleur calage, demandent la plupart du temps un nombre d'évaluations beaucoup plus important que les autres méthodes et sont difficilement utilisables pour des cas non synthétiques de réservoir (dans la plupart des cas, elles sont mêmes trop coûteuses pour les cas synthétiques). On se concentrera ainsi dans ce manuscrit sur les méthodes locales. Plus précisément, on décrira plus en détail les méthodes de descente et les méthodes à région de confiance avant de proposer une nouvelle méthode adaptée aux caractéristiques de la fonction objectif du problème de calage d'historique.

2.1.1 Méthodes de descente

Les méthodes présentées ici utilisent une information locale sur la fonction objectif qui garantit la plupart du temps, sous des hypothèses de régularité, une convergence de l'algorithme mais n'offre aucune assurance quant au caractère global de l'optimum trouvé. Les hypothèses de régularité peuvent être difficile à vérifier dans le cas du problème de calage d'historique. Par exemple, la condition de différentiabilité de la fonction objectif peut être très contraignante et entraîner des problèmes de convergence dans des cas réels.

Les algorithmes décrits dans cette section consistent à donner une estimation initiale de l'optimum puis à la mettre à jour en se déplaçant dans une direction de descente, comme illustré ci-dessous :

1. Choisir une estimation initiale de l'optimum x_0 et une tolérance ϵ.
2. À l'itération $k \geq 0$:
   (a) Trouver une direction de descente d_k.
   (b) Mettre à jour l'estimation de l'optimum, $x_{k+1} = x_k + \alpha_k d_k$ avec un pas α_k bien choisi.
3. Conditions d'arrêt (par exemple) : $\frac{\|x_{k+1} - x_k\|}{\|x_0\|} < \epsilon$.

La direction de descente d_k est en général calculée à partir des dérivées au premier et second ordre de la fonction objectif. Un exemple simple est de prendre $d_k = -\nabla F(x_k)$, auquel cas d_k est appelée la direction de plus forte pente (ou steepest descent). Les méthodes de type quasi Newton s'appuient quant à elles sur une estimation de la Hessienne de la fonction objectif en prenant $d_k = -\tilde{H}_k^{-1} \nabla F(x_k)$.

Il est donc important de calculer des approximations fiables des dérivées aux ordres un et deux de la fonction objectif. On présente ci-dessous cinq approches possibles de ce problème, pour la plupart très coûteuses en terme de nombre de simulations.

(i) Approximation du gradient par méthode adjointe

Une méthode courante en problèmes inverses pour calculer le gradient de la fonction objectif consiste à construire un problème adjoint au problème direct et, à partir de la résolution de ces deux problèmes d'obtenir une évaluation du gradient. Les domaines d'application peuvent être très vastes : Oberail et al. [76] utilisent par exemple ce type de méthode dans le milieu de l'imagerie médicale, Jarny et al. [57] l'utilisent pour résoudre des problèmes de conduction tandis que Fisher et al. [40] en présentent une application dans le domaine de l'optimisation de forme. Une telle méthode s'avère très efficace puisqu'une simple simulation du système et une résolution du problème adjoint permettent d'obtenir une bonne estimation du gradient. Cependant, il n'existe pas de méthode systématique pour construire le problème adjoint ; la construction d'un tel problème peut ainsi se révéler très délicate et nécessite une connaissance approfondie du problème direct.

Dans les problèmes inverses issus de l'ingénierie pétrolière, le problème direct correspond à la simulation des écoulements dans le réservoir [78]. Lorsqu'un problème adjoint est disponible, cette méthode offre d'excellents résultats [73, 51, 66, 96]. Cependant, les simulateurs industriels étant très complexes, il est impossible de produire un code adjoint dans la plupart des cas. Il est donc nécessaire de disposer d'une méthode alternative pour calculer le gradient pour le calage d'historique.

(ii) Approximation du gradient par différences finies

On peut estimer le gradient de F en approximant chacune de ses composantes par différences finies décentrées [75] :

$$\frac{\partial F(x)}{\partial x_i} \approx \frac{F(x + h_i e_i) - F(x)}{h_i}$$

où e_i est le $i^{\text{ème}}$ vecteur de la base canonique de $\mathbb{R}^p$.

Cette méthode d'approximation du gradient a notamment été utilisée pour des problèmes pétroliers dans [11, 49, 105]. Elle requiert par contre $(p+1)$ évaluations de la fonction objectif, ce qui devient très coûteux lorsque la dimension du problème augmente, phénomène mis en évidence dans [5] par exemple.

De plus, un problème récurrent avec de telles approximations consiste à choisir correctement le paramètre de pas h_i. En effet, le développement de Taylor de F d'où est issue la formule de différences finies nous incite à choisir h_i le plus petit possible. Toutefois, dans le cas où une erreur e est commise dans l'évaluation de F (ce qui est le cas dans un problème réel de simulation de réservoir), cette erreur se répercute en $\frac{e}{h_i}$ dans le calcul de $\frac{\partial F(x)}{\partial x_i}$. Une petite erreur peut alors avoir un très grand impact sur l'évaluation de ∇F. Le calibrage du paramètre h_i doit alors se faire à partir d'une estimation de l'erreur commise lors de l'évaluation de la fonction objectif, ce qui peut s'avérer très délicat dans la pratique lorsque les problèmes sont bruités.

(iii) Gradient stochastique - SPSA

Une méthode proposée par Spall dans [101, 102] permet de calculer une approximation du gradient d'une fonction à l'aide de perturbations aléatoires. Le gradient stochastique (ou SPSA) basique s'écrit :

$$\nabla F(x) \approx \frac{F(x + \epsilon\delta) - F(x)}{\epsilon\delta}$$

où $\delta = (\delta_1, \ldots, \delta_p)$ représente le vecteur de perturbation et $\epsilon > 0$ est l'amplitude de la perturbation. La notation avec δ en dénominateur correspond ici à une division terme à terme de

vecteurs, c'est à dire que la coordonnées i du gradient estimé de F est :

$$(\nabla F(x))_i \approx \frac{F(x+\epsilon\delta) - F(x)}{\epsilon\delta_i}$$

La $i^{\text{ème}}$ coordonnée de la perturbation δ est choisie dans la méthode de Spall en faisant un tirage aléatoire à partir d'une loi de Bernouilli symétrique ± 1, mais d'autres résultats [65] ont montré qu'il peut être avantageux de choisir une autre loi. A noter que l'écriture avec δ_i en dénominateur interdit de choisir une loi qui donne la possibilité de tirer une des coordonnées de δ à 0, l'une des hypothèses nécessaires étant même de choisir $\frac{1}{\delta_i}$ uniformément borné.

Deux propriétés importantes sont par ailleurs à signaler pour cette estimation du gradient :
— elle propose toujours une direction de descente, en l'occurrence la direction δ,
— son espérance est égale au gradient réel de la fonction objectif.
la première propriété permet d'assurer la convergence d'un algorithme de descente utilisant ce gradient. La seconde permet d'affiner grandement l'approximation du gradient en moyennant plusieurs estimations calculées à partir de différents tirages de δ.

Le principal intérêt de cette méthode est le faible coût pour avoir une estimation du gradient puisque deux évaluations suffisent à l'obtenir. Elle a d'ailleurs été utilisée dans [43, 65] pour des problèmes de calage d'historique, dans [38, 80, 115] pour des problèmes de maximisation de la production et dans [5, 4] pour l'optimisation du placement de puits. Malheureusement, le manque de précision dans ce calcul ralentit souvent beaucoup la vitesse de convergence des algorithmes d'optimisation, particulièrement dans le das où les problèmes traités sont de surcroit bruités.

(iv) Approximation de la Hessienne de la fonction objectif

Pour des raisons de coût de calcul, il n'est en général pas possible de calculer les dérivées d'ordre 2 de la fonction objectif ($\mathcal{O}(p^2)$ simulations). On utilise donc souvent des méthodes de mise à jour séquentielle de la Hessienne de la fonction objectif à partir de calculs effectués pour le gradient. L'algorithme de descente ainsi conçu est appelé algorithme de Quasi-Newton [29, 75, 99] : il s'agit d'une méthode de Newton qui n'utilise pas la valeur exacte de la Hessienne mais une approximation qui s'affine au cours des itérations. L'idée consiste alors à construire la Hessienne H_{k+1} à partir des valeurs de H_k, ∇F^k, ∇F^{k+1}, x_k et x_{k+1}. L'approximation BFGS (Broyden-Fletcher-GoldFarb-Shanno) [14, 41, 47, 99] consiste, en notant :

$$\begin{cases} \delta_k = x_{k+1} - x_k \\ \gamma_k = \nabla F^{k+1} - \nabla F^k \end{cases}$$

à mettre à jour H_{k+1} telle que :

$$H_{k+1} = H_k + \frac{\gamma_k \gamma_k^T}{\gamma_k^T \delta_k} - \frac{H_k \delta_k \delta_k^T H_k^T}{\delta_k^T H_k^T \delta_k}$$

Cette façon de mettre à jour la Hessienne à chaque itération de l'algorithme garantit que si H_k est symétrique définie positive, alors H_{k+1} le sera aussi. On initialise en général H_0 à un multiple de l'identité, de façon à maintenir ces propriétés tout au long de l'algorithme.

Une alternative plus simple peut être utilisée lorsque la fonction objectif s'écrit sous la forme d'une somme de résidus au carré [12, 64] (ce qui est le cas dans le problème de calage

d'historique) :

$$F(x_1, \ldots, x_p) = \sum_{i=1}^{n} r_i^2(x_1, \ldots, x_p)$$

Dans ce cas, les dérivées d'ordre 1 et 2 peuvent s'exprimer en fonction sous la forme suivante, lorsque la matrice Jacobienne s'écrit :

$$J(x) = \begin{bmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_n(x)^T \end{bmatrix}$$

Le gradient devient :

$$\nabla F(x) = \sum_{i=1}^{n} r_i(x) \nabla r_i(x) = J(x)^T r(x)$$

La Hessienne devient quant à elle :

$$\nabla^2 F(x) = J(x)^T J(x) + \sum_{i=1}^{n} r_j(x) \nabla^2 r_i(x)$$

L'approximation de Gauss-Newton consiste à négliger le second terme dans l'expression de la Hessienne : $\nabla^2 F(x) \approx J(x)^T J(x)$. Dans la plupart des cas, le premier terme $J(x)^T J(x)$ est prépondérant par rapport au second terme (particulièrement à l'approche de l'optimum x^*). Cette approximation offre l'avantage de ne pas nécessiter de calculs supplémentaires une fois que le calcul du gradient est effectué et d'approcher généralement beaucoup mieux la Hessienne que la méthode BFGS.

De telles approximations offrent de bons résultats pour les algorithmes de descente du deuxième ordre sans nécessiter d'évaluation supplémentaire par rapport à l'estimation du gradient. Des applications de ces méthodes dans l'ingénierie pétrolière peuvent être trouvées dans [44, 100].

(v) Approximation du gradient d'une fonction partiellement séparable

Les travaux de D. Ding et al. dans [36] ont montré qu'il est possible de réduire le coût en terme d'évaluations de la fonction objectif du calcul du gradient d'une fonction partiellement séparable. Plus précisément, on cherche à calculer le gradient d'une fonction de la forme :

$$F(x_1, \ldots, x_p) = \sum_{i=1}^{n} f_i(x_{i,1}, \ldots, x_{i,p_i})$$

où chaque sous-fonction f_i ne dépend que d'un nombre restreint de paramètres (pour tout i, $p_i < p$). Les dérivées partielles de F s'écrivent alors :

$$\frac{\partial F(x_1, \ldots, x_p)}{\partial x_j} = \sum_{i=1}^{n} \frac{\partial f_i(x_{i,1}, \ldots, x_{i,p_i})}{\partial x_j}$$

Notons p_{max} le nombre maximum de paramètres dont dépendent les f_i. On veut pouvoir calculer simultanément les dérivées partielles de toutes les sous-fonctions objectifs avec seulement L

perturbations de F ($p_{max} \leq L \leq p$ où p est le nombre de paramètres de F). La matrice des perturbations s'écrit :

$$A = \begin{bmatrix} a_{11} & \dots & a_{1p} \\ a_{21} & \dots & a_{2p} \\ \vdots & \ddots & \vdots \\ a_{L1} & \dots & a_{Lp} \end{bmatrix}$$

On peut remarquer que l'approximation du gradient par différences finies correspond au cas où $L = p$ et $A = Id$.

Pour tout $k \in [\![1, \dots, n]\!]$, et pour tout $i \in [\![1, \dots, L]\!]$, on note $B_k = (b_1^k, ..., b_L^k)$ le vecteur des différences de la sous-fonction objectif f_k entre les points perturbés P_i et le point courant P_0 : $b_i^k = f_k(P_i) - f_k(P_0)$. Les dérivées partielles d'une sous-fonction f_k peuvent alors être obtenues en résolvant le problème d'optimisation suivant :

$$\text{Trouver } Y_k^* = argmin J(Y_k)$$

avec

$$J(Y_k) = \frac{1}{2} \|A_k Y_k - B_k\|^2$$

et où A_k correspond à la matrice de perturbations associée à la fonction f_k :

$$A_k = \begin{bmatrix} a_{1,1_k} & \dots & a_{1,p_k} \\ a_{2,1_k} & \dots & a_{2,p_k} \\ \vdots & \ddots & \vdots \\ a_{L,1_k} & \dots & a_{L,p_k} \end{bmatrix}$$

Comme on a choisi $L \geq p_{max}$, la solution du problème aux moindres carrés donne une approximation du gradient de f_k à la seule condition que A_k soit de rang maximal. Pour obtenir une approximation du gradient de F, il suffit donc juste de choisir A de telle sorte que toutes les matrices A_k soient de rang maximal. Une bonne manière de la choisir est de la prendre localement de rang maximal (toutes les sous-matrices sont de rang maximal). Généralement, une matrice dont les éléments ont été générés aléatoirement présente cette caractéristique. Il est donc possible de calculer une approximation du gradient d'une fonction partiellement séparable avec seulement $L = p_{max}$ simulations, ce qui peut grandement améliorer les performances d'un algorithme d'optimisation basé sur le calcul du gradient.

Le choix des perturbations reste cependant problématique. En effet, le bon choix de ces perturbations est en lien direct avec la qualité de l'approximation du gradient. Une idée proposée dans [36] serait de trouver un ensemble de perturbations "optimal" en minimisant l'erreur commise sur l'approximation du gradient d'une fonction connue. Il est cependant difficile de garantir que l'approximation du gradient de la véritable fonction objectif restera pertinente.

2.1.2 Méthodes de recherche directe

Les méthodes de recherche directe consistent à échantillonner la fonction objectif sur un certain nombre de points et à décider des évaluations suivantes à réaliser en utilisant uniquement cet échantillonage. Elles ne nécessitent donc pas de construction de modèle de la fonction objectif ni d'approximation de ses dérivées. On présente deux types de méthodes : la première est basée sur la recherche de directions de descente tandis que la seconde considère l'évaluation de nouveaux points en fonction de la forme de simplexes. Ces méthodes présentent l'avantage de ne pas nécessiter la connaissance des dérivées de la fonction objectif tout en étant très simples à implémenter.

(i) Recherche directe directionnelle

On commence par donner la définition d'une base positive de $\mathbb{R}^n$ [27, 30].

Définition 1. — *Le cône positif engendré (positive span) par l'ensemble de vecteurs* $\{v_1, ..., v_r\}$ *de* $\mathbb{R}^n$ *est le cône convexe :*

$$\{v \in \mathbb{R}^n \text{ tel qu'on peut écrire } v = \alpha_1 v_1 + ... + \alpha_r v_r \text{ avec } \forall i, \alpha_i \geq 0\}. \tag{2.3}$$

— *Un ensemble engendrant* $\mathbb{R}^n$ *positivement est un ensemble de vecteurs de* $\mathbb{R}^n$ *dont le cône positif engendré est* $\mathbb{R}^n$*.*

— *Un ensemble de vecteurs est dit positivement dépendant si l'un des vecteurs de l'ensemble appartient au cône positif engendré par les autres.*

— *Une base positive de* $\mathbb{R}^n$ *est un ensemble de vecteurs positivement indépendants engendrant positivement* $\mathbb{R}^n$*.*

Les algorithmes de recherche directe directionnelle suivent alors la forme générale décrite dans l'algorithme 1.

Algorithme 1 Algorithme général de recherche directe

1. **Initialisation**
   — Choisir un point x_0 et des réels $\alpha_0 > 0$, $0 < \beta_1 \leq \beta_2 < 1$, $\gamma > 1$ et $\epsilon > 0$.
   — Choisir $\mathcal{D}$ un ensemble fini de bases positives de $\mathbb{R}^n$.
2. **Itération** k
   (a) **Exploration**
   — Choisir une base positive $\mathcal{D}_k \in \mathcal{D}$ et ordonner l'ensemble d'exploration $P_k = \{x_k + \alpha_k d, \text{ où } d \in \mathcal{D}_k\}$.
   — Évaluer la fonction objectif aux points de l'ensemble d'exploration dans l'ordre établi.
   — Si l'un des points d'exploration est tel que

   $$F(x_k + \alpha_k d) < F(x_k)$$

   alors l'itération est fructueuse : arrêter l'exploration et choisir $x_{k+1} = x_k + \alpha_k d$.
   (b) **Mise à jour du pas d'exploration** α_k
   — Si l'itération a été fructueuse, alors augmenter le pas d'exploration : $\alpha_{k+1} \in [\alpha_k, \gamma\alpha_k]$.
   — Sinon, réduire le pas d'exploration : $\alpha_{k+1} \in [\beta_1\alpha_k, \beta_2\alpha_k]$.
   (c) **Test d'arrêt** Arrêter l'algorithme lorsque le pas d'exploration devient trop petit ($\alpha_k < \epsilon$).

Le fait de choisir des bases positives de $\mathbb{R}^n$ permet de garantir qu'il est possible d'explorer l'espace dans son ensemble avec les points d'exploration définis dans l'étape 2(a) de l'algorithme 1. Des informations plus détaillées peuvent être trouvées dans [2, 106]. Kolda et al. [60] ont par ailleurs publié en 2003 une revue des méthodes classiques de ce type.

On peut par exemple considérer le cas de la base positive canonique :

$$\mathcal{D} = [e_1, ..., e_n, -e_1, ... - e_n] \tag{2.4}$$

Le cas le plus simple pour l'algorithme 1 consiste alors à choisir uniquement dans $\mathcal{D}$ la base positive canonique, l'algorithme résultant est appelé recherche directe par coordonnées.

(ii) Recherche directe par simplexe
L'algorithme Nelder-Mead [74] est un algorithme particulier classique de recherche directe. Chaque itération détermine l'itéré suivant en fonction de la valeur de la fonction objectif aux $n+1$ sommets d'un simplexe $Y = \{y_0, y_1, ..., y_n\}$ ordonnés dans un ordre croissant (suivant la fonction objectif à optimiser). Quatre opérations de base fondées sur le centre de gravité $y_g = \sum_{i=0}^{n-1} y_i$ du meilleur simplexe à n sommets sont possibles dans cet algorithme. Elles consistent toutes à calculer un nouveau point y de la forme

$$y = y_g + \delta(y_g - y_n)$$

où δ est un réel. Dans l'algorithme Nelder-Mead,

1. l'opération correspond à une réflexion lorsque $\delta = \delta_r = 1$
2. l'opération correspond à une expansion lorsque $\delta = \delta_e = 2$
3. l'opération correspond à une contraction extérieure lorsque $\delta = \delta_{ce} = 0.5$
4. l'opération correspond à une contraction intérieure lorsque $\delta = \delta_{ci} = -0.5$

Ces coefficients correspondent aux choix standards. Dans un cadre plus général on peut simplement choisir $-1 < \delta_{ci} < 0 < \delta_{ce} < \delta_r < \delta_e$.
L'algorithme a aussi la possibilité d'effectuer un rétrécissement dans le cas où aucune des opérations précédentes n'a donné de point correct. Un rétrécissement consiste à supprimer tous les points du simplexe Y excepté y_0 et à calculer pour tout $i \in \{1, ...n\}$

$$y_i^{ret} = y_0 + \gamma(y_i - y_0)$$

avec $0 < \gamma < 1$. Le nouveau simplexe devient alors $Y^{ret} = \{y_0, y_1^{ret}..., y_n^{ret}\}$. On présente dans l'algorithme 2 les détails de la méthode Nelder-Mead.
L'algorithme Nelder-Mead est populaire grâce à la fois à sa simplicité et à sa capacité à prendre en compte la courbure de la fonction objectif. En l'état, il existe des cas dans lesquels il ne convergera pas vers des points critiques de la fonction objectif. Il est cependant possible de l'adapter pour qu'il converge globalement vers des points critiques [61, 71].

Les algorithmes de recherche directe sont très simples à implémenter et à utiliser (ils nécessitent très peu de paramètres à calibrer). Cependant, leur efficacité n'est souvent pas la meilleure et ils deviennent très vite trop coûteux pour nos applications. On peut néanmoins trouver des applications de ces méthodes à des problèmes pétroliers dans [8, 20, 21]. Des arguments pour l'utilisation de ce type de méthode peuvent être avancés lorsqu'il est possible d'effectuer en parallèle les simulations d'écoulement. L'article [8] en particulier montre que lorsque le nombre de simulations effectuées en parallèle augmente, les méthodes de recherche directe ne donnent pas forcément de moins bons résultats que les méthodes de type SQP, "Sequential Quadratic Programming" présentés brièvement dans la section suivante.

Algorithme 2 Algorithme Nelder-Mead

1. **Initialisation**
   — Choisir un simplexe initial $Y_0 = \{y_0^0, y_1^0, ..., y_n^0\}$.
   — Évaluer F aux points de Y_0.
   — Choisir les constantes $0 < \gamma < 1$ et $-1 < \delta_{ci} < 0 < \delta_{ce} < \delta_r < \delta_e$.
2. **Itération** k
   Fixer $Y = Y_k$ et calculer $y_g = \sum_{i=0}^{n-1} y_i$.
   (a) **Ordonner les sommets de** Y
   Classer les sommets de Y de telle sorte que
   $$F(y_0) \leq F(y_1) \leq ... \leq F(y_n)$$
   (b) **Réflexion**
   — Calculer y_r la réflexion de y_n par y_g.
   — Évaluer $F(y_r)$.
   — Si $F(y_0) \leq F(y_r) < F(y_{n-1})$, alors remplacer y_n par y_r dans Y et terminer l'itération : $Y_{k+1} = \{y_0, y_1, ...y_{n-1}, y_r\}$.
   (c) **Expansion**
   Si $F(y_r) < F(y_0)$, alors
   — Évaluer y_e le point étendu.
   — Si $F(y_e) \leq F(y_r)$, alors remplacer y_n par y_e dans Y et terminer l'itération : $Y_{k+1} = \{y_0, y_1, ...y_{n-1}, y_e\}$.
   — Si $F(y_r) < F(y_e)$, alors remplacer y_n par y_r dans Y et terminer l'itération : $Y_{k+1} = \{y_0, y_1, ...y_{n-1}, y_r\}$.
   (d) **Contraction extérieure**
   Si $F(y_r) < F(y_n)$, alors
   — Évaluer y_{ce} le contracté extérieur.
   — Si $F(y_{ce}) \leq F(y_r)$, alors remplacer y_n par y_{ce} dans Y et terminer l'itération : $Y_{k+1} = \{y_0, y_1, ...y_{n-1}, y_{ce}\}$.
   — Sinon, effectuer un rétrécissement du simplexe (étape 2.(f)).
   (e) **Contraction intérieure**
   Si $F(y_r) \geq F(y_n)$, alors
   — Évaluer y_{ci} le contracté intérieur.
   — Si $F(y_{ci}) \leq F(y_n)$, alors remplacer y_n par y_{ci} dans Y et terminer l'itération : $Y_{k+1} = \{y_0, y_1, ...y_{n-1}, y_{ci}\}$.
   — Sinon, effectuer un rétrécissement du simplexe (étape 2.(f)).
   (f) **Rétrécissement du simplexe**
   $Y_{k+1} = \{y_0 + \gamma(y_i - y_0), \text{ pour } i = 0, ..., n\}$.

2.1.3 Méthodes de type région de confiance

Ces méthodes consistent à construire des modèles quadratiques successifs de la fonction objectif dont l'optimisation sur une région dite de "confiance" est simple et peut être appelée avec un coût marginal. C'est particulièrement le cas pour les problèmes pétroliers puisque l'optimisation de tels modèles quadratiques prend au maximum quelques secondes tandis que l'évaluation de la fonction objectif demande plusieurs heures de temps CPU.

Un algorithme de région de confiance s'écrit souvent sous la forme [25, 27, 75] de l'algorithme 3

Algorithme 3 Un algorithme de type région de confiance

1. **Initialisation**
   Choisir x_0 un point, $\Delta_0 > 0$ une taille de région de confiance autour de x_0 et construire un modèle quadratique initial m_0 de la fonction objectif.
2. **Itération k**
   — S'assurer de la correction du modèle
   — Lorsque le modèle est correct, calculer $x_k^+ = \min_{x \in B(x_k, \Delta_k)} m_k(x)$
   — Mettre à jour le modèle en évaluant F au point x_k^+
   — Si x_k^+ est un meilleur point que x_k, définir $x_{k+1} = x_k^+$, sinon $x_{k+1} = x_k$
3. **Condition d'arrêt**
   Arrêter l'algorithme lorsque le gradient du modèle m_k devient trop petit.

Plusieurs méthodes sont applicables pour calculer les modèles quadratiques, on en présente brièvement deux ci-dessous.

(i) Estimation du gradient et de la Hessienne de F
L'idée ici est d'utiliser des approximations du gradient et de la Hessienne de la fonction objectif pour construire les modèles quadratiques. Le modèle m_k à l'itération k de l'algorithme à région de confiance s'écrit alors simplement :

$$m_k(x_k + x) = F(x_k) + \nabla F(x_k)^T x + \frac{1}{2} x^T \nabla^2_{xx} F(x_k) x$$

Ce type de méthode à région de confiance fait partie des méthodes SQP (Sequential Quadratic Programming) [15, 16, 108]. Une revue générale de la littérature dans le domaine a été effectuée par Gould et al. [50]. Typiquement, ce type d'algorithmes produit de bons résultats dans les cas pétroliers [46, 110]. Ces méthodes fournissent d'excellents résultats lorsque les gradients et Hessiennes de la fonction objectif sont bons. Néanmoins, les mêmes problèmes de coût et de précision de l'estimation de ces quantités que pour les algorithmes de descente peuvent survenir en l'absence de problème adjoint.

(ii) Interpolation quadratique
Afin de s'affranchir des problèmes liés aux évaluations des dérivées de la fonction objectif, on peut créer les modèles en interpolant F sur des points correctement choisis. Ce type d'optimisation sans dérivées donne souvent de très bons résultats sur des problèmes pétroliers [116, 63, 114]. C'est sur ce type de méthodes que nous avons choisi de travailler en raison de ses qualités et du fait qu'elles ne nécessitent pas d'évaluation de dérivées. Nous les décrivons avec beaucoup plus de détails dans la section suivante.

2.2 Description du fonctionnement des méthodes de type région de confiance avec interpolation quadratique

2.2.1 Un algorithme simpliste

Dans une optique d'optimisation par une méthode de type région de confiance avec des modèles d'interpolation quadratique, l'algorithme simpliste 4 est l'un de ceux pouvant être conçu intuitivement.

Algorithme 4 Un algorithme simpliste de type région de confiance avec modèle d'interpolation

1. **Initialisation**
   - Choisir un point x_0 et un rayon de région de confiance Δ_0 initiaux.
   - Sélectionner un ensemble d'interpolation initial Y_0 contenant q points dont x_0 et évaluer F en les points de Y_0.
   - Construire le modèle initial m_0 qui interpole F aux points de Y_0.
   - Fixer les constantes $\epsilon > 0$ et $0 < \gamma_{low} < 1$.
2. **Itération** k
   (a) **Test d'arrêt**
   - Calculer le gradient du modèle au point courant x_k : $g_k = \nabla m_k(x_k)$
   - Si $\|g_k\| \leq \epsilon$, arrêter l'algorithme, sinon passer à l'étape suivante

   (b) **Calcul du nouveau point**
   Calculer un point x_k^+ de telle sorte que :
   $$m_k(x_k^+) = \min_{x \in B(x_k, \Delta_k)} m_k(x)$$
   Évaluer $F(x_k^+)$
   i. **Itération fructueuse :** Si $F(x_k^+) < F(x_k)$, remplacer un point de Y_k par x_k^+ et choisir $x_{k+1} = x_k^+$.
   ii. **Itération infructueuse :** Si $F(x_k^+) \geq F(x_k)$, choisir $x_{k+1} = x_k$.
   iii. **Gestion de la région de confiance :**
   - Si $x_{k+1} = x_k$, choisir $\Delta_{k+1} = \gamma_{low}\Delta_k$
   - Si $x_{k+1} = x_k^+$, choisir $\Delta_{k+1} \geq \Delta_k$

   (c) **Mise à jour du modèle**
   Calculer le nouveau modèle d'interpolation m_{k+1} centré sur x_{k+1} et utilisant Y_{k+1}. Retourner ensuite à l'étape 2.

Un tel algorithme peut fonctionner correctement et parfois donner de très bons résultats numériques, comme présenté dans l'article de Fasano et al. [39]. Cependant, Scheinberg et Toint ont montré dans [98] qu'il n'est pas possible d'ignorer complètement la qualité de la géométrie des ensembles d'interpolation pour assurer la convergence de l'algorithme. En effet, dans l'algorithme 4, l'obtention d'un mauvais point d'essai x_k^+ peut être due à deux causes distinctes :

- soit la région de confiance est trop grande pour qu'un modèle quadratique puisse être une bonne approximation de F ;
- soit la géométrie de l'ensemble d'interpolation n'est pas assez bonne pour construire un modèle quadratique approchant correctement F.

En l'état, seule la première cause est prise en compte dans l'algorithme au-dessus ; il n'y a en effet aucun moyen d'améliorer la géométrie des ensembles d'interpolation si cela devient nécessaire.

La figure 2.1 issue de [98] montre un cas en dimension 2 pour lequel l'algorithme tend à n'ajouter que des points sur une unique droite.

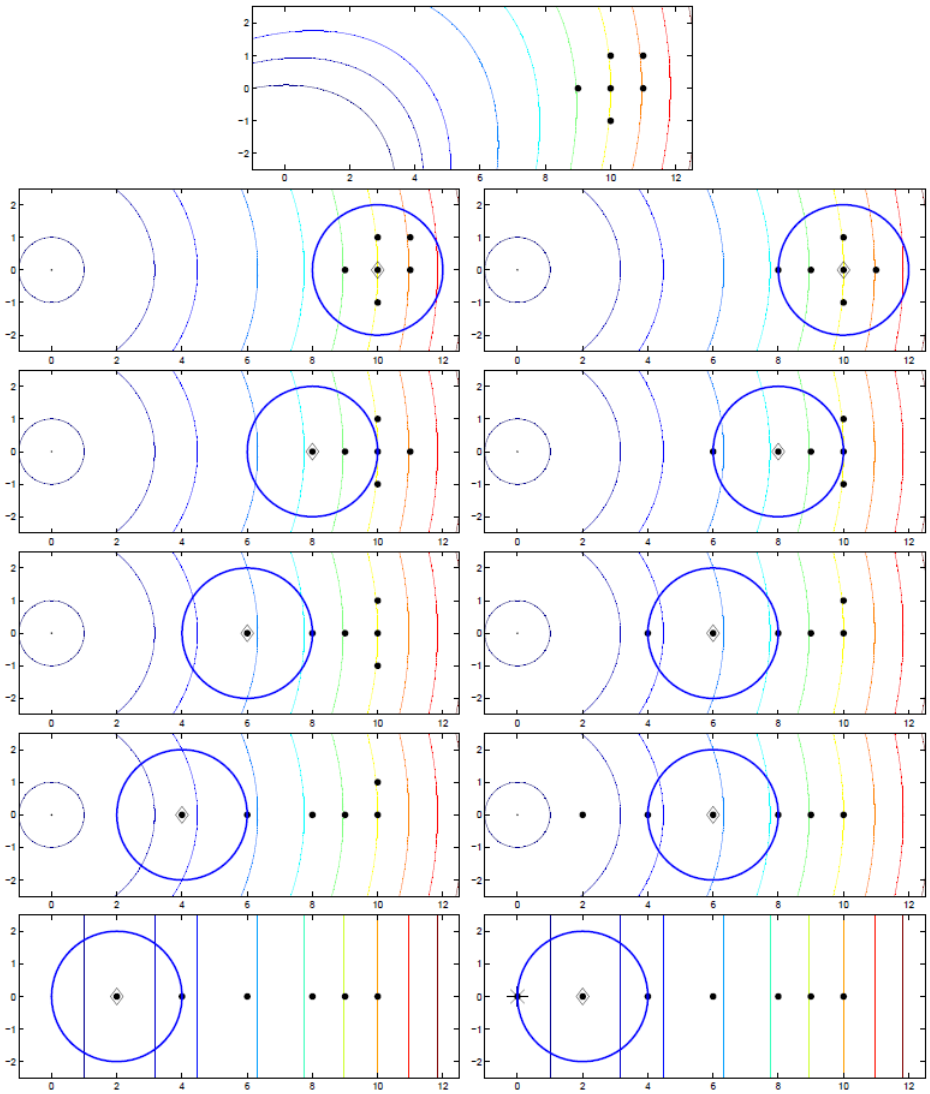


FIGURE 2.1 – Exemple d'une fonction et d'un ensemble d'interpolation qui mène l'algorithme à construire des modèles dégénérés (issu de [98]).

Les modèles m_k successifs n'ont alors plus aucune chance d'être de bonnes approximations de la fonction objectif quelle que soit la taille des régions de confiance et mènent l'algorithme à s'arrêter sur des points non stationnaires. Il est donc nécessaire pour un algorithme de type région de confiance utilisant des modèles d'interpolation de disposer d'un moyen de garantir la qualité de la géométrie des ensembles de points d'interpolation. On présentera dans les sous-sections suivantes une mesure de la qualité des ensembles d'interpolation ainsi qu'une procédure séquentielle permettant d'améliorer ces géométries.

2.2.2 Construction des modèles quadratiques

Deux méthodes sont possibles ici et présentées ci-dessous.

2.2.2.1 Décomposition dans une base de polynômes quadratiques

Pour construire un modèle d'interpolation quadratique de F, on commence par choisir une base $\phi = \{\phi_1, ..., \phi_{L_p}\}$ de $\mathcal{P}_2^p$ (espace des polynômes à p variables de degré ≤ 2), où $L_p = \frac{(p+1)(p+2)}{2}$ est la dimension de $\mathcal{P}_2^p$. On aura donc besoin d'évaluer F en L_p points $Y = \{y_1, \ldots, y_{L_p}\}$ de $\mathbb{R}^p$ pour être capable de construire un modèle d'interpolation $m \in \mathcal{P}_2^p$ unique. On définit :

$$M(\phi, Y) = \begin{bmatrix} \phi_1(y_1) & \phi_2(y_1) & \cdots & \phi_{L_p}(y_1) \\ \phi_1(y_2) & \phi_2(y_2) & \cdots & \phi_{L_p}(y_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1(y_{L_p}) & \phi_2(y_{L_p}) & \cdots & \phi_{L_p}(y_{L_p}) \end{bmatrix}$$
$$\alpha_\phi = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{L_p} \end{bmatrix} \text{ et } F(Y) = \begin{bmatrix} F(y_1) \\ F(y_2) \\ \vdots \\ F(y_{L_p}) \end{bmatrix}. \tag{2.5}$$

Les coordonnées α_ϕ dans ϕ du modèle m interpolant F aux points de Y sont solution du problème linéaire :

$$M(\phi, Y)\alpha_\phi = F(Y) \tag{2.6}$$

m existe et est unique si et seulement si la matrice $M(\phi, Y)$ n'est pas singulière, c'est à dire si tous les points de Y ne résident pas sur une même variété quadratique. On définit pour cela la notion d'unisolvance :

Définition 2 (unisolvance)**.** *Soient Y un ensemble de L_p points d'interpolation dans $\mathbb{R}^p$ et ϕ une base de $\mathcal{P}_2^p$. On dit que Y est unisolvant ("poised" en anglais) si la matrice $M(\phi, Y)$ n'est pas singulière.*

2.2.2.2 Calcul des polynômes de Lagrange

Une autre façon équivalente de construire un modèle d'interpolation quadratique est de considérer des bases de polynômes de Lagrange.

Définition 3. *Une base $\{l_1, ..., l_{L_p}\}$ de $\mathcal{P}_2^p$ est une base de Lagrange pour l'ensemble de points $Y = \{y_1, \ldots, y_{L_p}\}$ si elle vérifie :*

$$\forall (i,j) \in [\![1, ..., L_p]\!]^2,\ l_j(y_i) = \delta_{ij} = \begin{cases} 1 \text{ si } i = j \\ 0 \text{ si } i \neq j \end{cases} \tag{2.7}$$

Chaque polynôme d'une base de Lagrange est un polynôme qui interpole une fonction qui vaut 1 sur un des points d'interpolation et s'annule sur les autres. Il en découle que, par définition de l'unisolvance, si un ensemble de points d'interpolation Y est unisolvant, la base de Lagrange qui lui est associée existe et est unique. L'unique modèle qui interpole une fonction F aux points de Y peut alors être construit simplement grâce à l'expression :

$$m(x) = \sum_{i=1}^{L_p} F(y_i) l_i(x) \tag{2.8}$$

De façon équivalente au paragraphe précédent, on peut montrer que l'ensemble d'interpolation Y est unisolvant si et seulement si la base de polynômes de Lagrange qui lui est associée existe et est unique.

Numériquement, la construction des polynômes de Lagrange nécessite la résolution d'un système linéaire pour chacun des L_p points de l'ensemble d'interpolation Y, c'est donc un processus d'une complexité en $\mathcal{O}(L_p^3)$. D'un autre côté, si les points de Y ne sont remplacés que un par un, il est possible de mettre à jour ces polynômes en un coût de l'ordre de $\mathcal{O}(L_p^2)$. Pour les applications dans lesquelles l'évaluation de la fonction objectif en un point est très long devant la résolution d'un système linéaire, ce coût de calcul est marginal. Lorsque ce n'est pas le cas, l'optimisation du code de calcul de ces polynômes de Lagrange est nécessaire pour assurer de bonnes performances des algorithmes d'optimisation.

2.2.3 Λ-unisolvance des ensembles d'interpolation

Dans un algorithme d'optimisation sans dérivées, il est nécessaire que les modèles d'interpolation conduisent à une bonne approximation de la fonction objectif sur un domaine fermé $B \subset \mathbb{R}^p$. On a montré dans la section précédente que pour pouvoir construire un modèle d'interpolation, l'ensemble des points d'interpolation Y devait être unisolvant. Pourtant, cette condition seule d'unisolvance ne donne aucune information sur la qualité du modèle sur B. On a donc besoin d'une façon de mesurer la qualité de la répartition des points de Y dans B. Une mesure fréquemment utilisée est celle de Λ-unisolvance.

Définition 4 (Λ-unisolvance)**.** *Soit B un sous-ensemble fermé de $\mathbb{R}^p$, $Y = \{y_1, y_2, \ldots, y_{L_p}\}$ un ensemble de points d'interpolation unisolvant, $\Lambda > 0$ et $\{l_1, \ldots, l_{L_p}\}$ les polynômes de Lagrange associés à Y. Y est dit Λ-unisolvant ("Λ-poised") si :*

$$\max_{1 \leq i \leq L_p} \max_{x \in B} |l_i(x)| \leq \Lambda$$

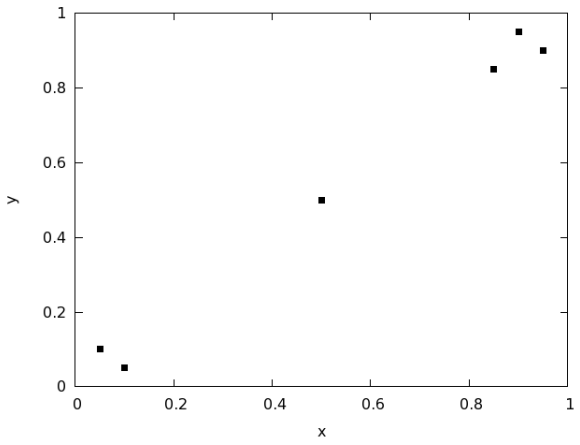
La Λ-unisolvance peut être interprétée comme une sorte de mesure d'inverse de la distance à un ensemble non-unisolvant ($\Lambda \geq 1$ et plus Λ est grand, plus l'ensemble d'interpolation est proche de la dépendance linéaire). La figure 2.2 montre par exemple deux ensembles de points unisolvants en dimension 2. Le premier est "presque" non-unisolvant (tous les points sont proches d'une même droite) alors que le second est bien réparti dans la boule $B(0, 1)$. La Λ-unisolvance donne une bonne indication sur la qualité de ces sous-ensembles. En effet, Le premier ensemble est Λ-unisolvant pour $\Lambda \geq 440$ tandis que le second l'est pour $\Lambda \geq 1$.

Il est intéressant de remarquer que cette définition n'impose pas de condition sur la position des points d'interpolation par rapport au domaine B. Cependant, plus ils en sont éloignés, plus la Λ-unisolvance de l'ensemble Y a des chances d'être élevée. C'est pourquoi dans la plupart des algorithmes d'optimisation fondés sur des modèles d'interpolation, on cherchera d'abord à remplacer les points éloignés de la région de confiance.

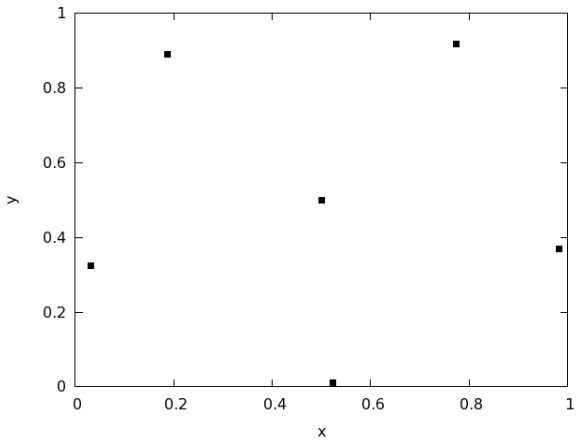
2.2.4 Modèles d'interpolation sous-déterminés

Les modèles construits avec l'une des méthodes de la section 2.2.2 sont faciles à déterminer et généralement assez précis sur une région contenant les points d'interpolation. En revanche, ils nécessitent l'évaluation de F en L_p points, ce qui est beaucoup trop coûteux pour être utilisé dans la pratique. Afin d'être capable d'utiliser nos processus d'optimisation, on a donc besoin de construire des modèles d'interpolation avec un nombre plus restreint de points $q < L_p$. Avec moins de L_p points et pour une base ϕ, la matrice $M(\phi, Y)$ comprend plus de colonnes que de lignes et le système (2.6) ne possède pas de solution unique. Une des possibilités est de choisir la solution α de norme euclidienne minimale. Si $M(\phi, Y)$ est de rang maximal, son pseudo-inverse existe et la solution de norme minimale est donnée par la formule :

$$\alpha = M(\phi, Y)^T [M(\phi, Y) M(\phi, Y)^T]^{-1} F(Y). \tag{2.9}$$



(a) Mauvaise Λ-unisolvance ($\Lambda \geq 440$)



(b) Bonne Λ-unisolvance ($\Lambda \geq 1$)

FIGURE 2.2 – Un ensemble d'interpolation présentant une mauvaise Λ-unisolvance (a) et un autre présentant un bonne Λ-unisolvance (b) (voir [27]).

Cette solution est robuste par rapport aux petites variations de l'ensemble d'interpolation. Elle est cependant dépendante du choix de la base ϕ. Il est donc possible d'obtenir de meilleurs ou de moins bons modèles suivant la base dans laquelle ils sont construits. Identifier la meilleure de ces bases pour un cas donné n'est pas aisé, le choix naturel se porte donc sur la base sur laquelle de nombreux résultats ont été montrés : la base canonique $\bar{\phi}$ de $\mathcal{P}_2^p$.

Dans une optique d'optimisation, il est souvent désirable de construire des modèles avec une partie linéaire précise puis d'ajouter progressivement l'information sur la courbure de la fonction objectif à l'aide de nouveaux points d'interpolation. Si $\bar{\phi} = \{1, x_1, \ldots, x_p, \frac{1}{2}x_1^2, x_1x_2, \ldots, \frac{1}{2}x_p^2\}$ est la base canonique de $\mathcal{P}_2^p$, $\bar{\phi}_L = \{1, \ldots, x_p\}$ sa partie linéaire et $\bar{\phi}_Q = \{\frac{1}{2}x_1^2, x_1x_2, \ldots, \frac{1}{2}x_p^2\}$ sa partie quadratique, on peut écrire le modèle d'interpolation recherché sous la forme :

$$m(x) = \alpha_L^T \bar{\phi}_L(x) + \alpha_Q^T \bar{\phi}_Q(x). \tag{2.10}$$

Si Y contient au moins $p+2$ points, on définit alors le modèle d'interpolation (α_L^*, α_Q^*) comme étant la solution du problème :

$$\begin{cases} (\alpha_L^*, \alpha_Q^*) = argmin \dfrac{1}{2} \|\alpha_Q\|^2 \\ \text{sous la contrainte : } M(\bar{\phi}_L, Y)\alpha_L + M(\bar{\phi}_Q, Y)\alpha_Q = F(Y). \end{cases} \tag{2.11}$$

On appelle la solution de ce problème, le modèle de norme de Frobenius minimale. En effet, en utilisant la base canonique $\bar{\phi}$ et en séparant la solution en $\alpha = (\alpha_L, \alpha_Q)$, minimiser $\|\alpha_Q\|$ est équivalent à minimiser la norme de Frobenius de la Hessienne du modèle m.

Le problème (2.11) possède une unique solution si et seulement si la matrice suivante est non singulière :

$$F(\bar{\phi}, Y) = \begin{bmatrix} M(\bar{\phi}_Q, Y)M(\bar{\phi}_Q, Y)^T & M(\bar{\phi}_L, Y) \\ M(\bar{\phi}_L, Y)^T & 0 \end{bmatrix}. \tag{2.12}$$

Si cette condition est vérifiée, Y est dit unisolvant au sens de la norme de Frobenius. Dans la suite du document, tous les modèles sous-déterminés considérés seront des modèles de norme de Frobenius minimale.

Comme pour le cas de modèles déterminés, on peut définir les polynômes de Lagrange au sens de la norme de Frobenius pour des ensembles Y contenant moins de L_p points : $\alpha_i^*(x) = \alpha_{L,i}^{*T}\Phi_L(x) + \alpha_{Q,i}^{*T}\Phi_Q(x)$, avec $(\alpha_{L,i}^*, \alpha_{Q,i}^*)$ solution du problème

$$\begin{cases} (\alpha_{L,i}^*, \alpha_{Q,i}^*) = argmin \frac{1}{2} \|\alpha_{Q,i}\|^2 \\ \text{sous la contrainte : } M(\bar{\phi}_L, Y)\alpha_{L,i} + M(\bar{\phi}_Q, Y)\alpha_{Q,i} = I_{y_i}(Y). \end{cases} \tag{2.13}$$

où I_{y_i} est la fonction indicatrice du point y_i. Chaque composante de la solution $\alpha_i = (\alpha_{L,i}, \alpha_{Q,i})$ interpole une fonction qui vaut 1 sur un des points de Y et 0 sur les autres. Cette propriété généralise bien les polynômes de Lagrange au cas sous-déterminé.

De la même façon que pour le cas entièrement déterminé, il est possible de définir les modèles d'interpolation de norme de Frobenius minimale à l'aide des polynômes de Lagrange :

$$m(x) = \sum_{i=1}^{q} F(y_i) l_i(x). \tag{2.14}$$

Ici encore, la construction des polynômes de Lagrange nécessite la résolution d'un système linéaire pour chaque point d'interpolation et est donc un processus de complexité $\mathcal{O}(q^3)$. Le processus de mise à jour de ces polynômes lorsqu'un point de Y est remplacé est quant à lui de complexité $\mathcal{O}(q^2)$.

On peut enfin définir de façon analogue au cas déterminé la notion de Λ-unisolvance :

Définition 5. *Soit B un sous-ensemble fermé de $\mathbb{R}^p$, $Y = \{y_1, y_2, \ldots, y_q\}$ un ensemble de points d'interpolation unisolvant au sens de la norme de Frobenius ($q \leq L_p$), $\Lambda > 0$ et $\{l_1, \ldots, l_q\}$ les polynômes de Lagrange associés à Y. Y est dit Λ-unisolvant (Λ-poised) si :*

$$\max_{1 \leq i \leq q} \max_{x \in B} |l_i(x)| \leq \Lambda$$

Comme auparavant, la Λ-unisolvance donne un moyen de mesurer la distance de l'ensemble d'interpolation Y à un ensemble non-unisolvant.

2.2.5 Initialisation des modèles quadratiques

Il est en général trop coûteux d'évaluer F en L_p points lors de la construction du premier modèle quadratique m_0 dans un algorithme sans dérivées du type de l'algorithme 4. C'est pourquoi on travaille souvent avec un modèle sous-déterminé de norme de Frobenius minimale. De bons résultats numériques sont généralement obtenus lorsqu'on choisit un nombre de points d'interpolation initial égal à $2p + 1$. Il est possible de commencer avec moins de points d'interpolation (deux points peuvent suffire pour débuter l'algorithme) mais le choix de $2p+1$ points nous a paru raisonnable pour garantir un bon modèle initial. Afin de maximiser la précision du modèle m_0, on cherche à répartir au mieux ces points dans la région de confiance. Plus précisément, étant donnés initialement un point x_0 et un rayon de région de confiance Δ_0, on choisit l'ensemble d'interpolation initial $Y = \{y_0, y_1, ..., y_{2p}\}$ comme suit :

$$y_0 = x_0 \text{ et } \forall i \in \{0, 1, ..., p-1\} \begin{cases} y_{i+1} = y_0 + \Delta_0 e_i \\ y_{i+p+1} = y_0 - \Delta_0 e_i \end{cases} \tag{2.15}$$

où e_i représente le $i^{\text{ème}}$ vecteur de la base canonique de $\mathbb{R}^p$.
Un tel ensemble d'interpolation est Λ-unisolvant dans $B(x_0, \Delta_0)$ pour $\Lambda = 1$.

2.2.6 Précision des modèles et amélioration de la Λ-unisolvance

2.2.6.1 Modèles entièrement linéaires

Pour assurer la convergence d'un algorithme de région de confiance, il est nécessaire de garantir que le rayon de la région de confiance ne tende pas vers 0 hors des points stationnaires. Si les modèles locaux sont construits à l'aide des dérivées de la fonction objectif, il est facile de montrer qu'une diminution de la taille de la région de confiance induit une amélioration de la précision du modèle avec la proposition suivante :

Proposition 2. *On suppose que la fonction objectif F est deux fois continûment dérivable avec une Hessienne continue et Lipschitzienne. Si on considère un modèle quadratique construit à partir des dérivées de F sur une boule $B(x_0, \Delta)$, alors il existe des constantes k_{eh}, k_{eg} et k_{ef} telles que*

— *l'erreur entre la Hessienne du modèle et celle de F vérifie*

$$\forall x \in B(x_0, \Delta),\ \left\|\nabla^2 F(x) - \nabla^2 m(x)\right\| \leq k_{eh}\Delta,$$

— *l'erreur entre le gradient du modèle et celui de F vérifie*

$$\forall x \in B(x_0, \Delta),\ \|\nabla F(x) - \nabla m(x)\| \leq k_{eg}\Delta^2,$$

— *l'erreur entre le modèle et F vérifie*

$$\forall x \in B(x_0, \Delta),\ \|F(x) - m(x)\| \leq k_{ef}\Delta^3.$$

Dans le cas où les modèles sont construits sans prendre en compte d'information sur la dérivée, on va chercher à avoir des propriétés similaires à la proposition 2. Pour cela, on commence par introduire la notion de modèles entièrement linéaires.

Définition 6. *Soient un ensemble $S \subset \mathbb{R}^p$ et un rayon maximal Δ_{max} donnés. On suppose que la fonction objectif F est continûment dérivable avec un gradient continu et Lipschitzien sur $\cup_{x\in S} B(x, \Delta_{max})$. Un ensemble de modèles $\mathcal{M}$ est appelé classe de modèles entièrement linéaire si :*

1. *Il existe des constantes k_{eg} et k_{ef} telles que pour tout $x_0 \in S$ et $\Delta \leq \Delta_{max}$, il existe un modèle m de $\mathcal{M}$ tel que*
   — *l'erreur entre le gradient du modèle et celui de F vérifie*

   $$\forall x \in B(x_0, \Delta),\ \|\nabla F(x) - \nabla m(x)\| \leq k_{eg}\Delta$$

   — *l'erreur entre le modèle et F vérifie*

   $$\forall x \in B(x_0, \Delta),\ \|F(x) - m(x)\| \leq k_{ef}\Delta^2.$$

   Un tel modèle m est dit entièrement linéaire sur $B(x_0, \Delta)$.
2. *Pour cette classe $\mathcal{M}$, il existe un algorithme d'amélioration de modèle qui peut, en un nombre fini d'étapes :*
   — *soit garantir qu'un modèle $m \in \mathcal{M}$ est entièrement linéaire sur $B(x_0, \Delta)$*
   — *soit trouver un modèle $\tilde{m} \in \mathcal{M}$ qui est entièrement linéaire sur $B(x_0, \Delta)$.*

Disposer d'une famille de modèles vérifiant cette définition est nécessaire pour les preuves de convergence d'algorithmes à région de confiance. On peut alors garantir qu'un algorithme utilisant des modèles entièrement linéaires converge vers un point critique du premier ordre. On peut de la même façon définir les modèles entièrement quadratiques.

Définition 7. *Soient un ensemble S et un rayon maximal Δ_{max} donnés. On suppose que la fonction objectif F est deux fois continûment dérivable avec une Hessienne continue Lipschitz sur $\cup_{x\in S} B(x, \Delta_{max})$. Un ensemble de modèle $\mathcal{M}$ est appelé classe de modèle entièrement quadratique si*

1. *Il existe des constantes k_{eg}, k_{ef} et k_{eh} telles que pour tout $x \in S$ et $\Delta \leq \Delta_{max}$, il existe un modèle m de $\mathcal{M}$ tel que*
   - *l'erreur entre la Hessienne du modèle et celle de F vérifie*

$$\forall x \in B(x_0, \Delta),\ \left\|\nabla^2 F(x) - \nabla^2 m(x)\right\| \leq k_{eh}\Delta$$

   - *l'erreur entre le gradient du modèle et celui de F vérifie*

$$\forall x \in B(x_0, \Delta),\ \|\nabla F(x) - \nabla m(x)\| \leq k_{eg}\Delta^2$$

   - *l'erreur entre le modèle et F vérifie*

$$\forall x \in B(x_0, \Delta),\ \|F(x) - m(x)\| \leq k_{ef}\Delta^3.$$

2. *Pour cette classe $\mathcal{M}$, il existe un algorithme d'amélioration de modèle qui peut en un nombre fini d'étapes*
   - *soit garantir qu'un modèle $m \in \mathcal{M}$ est entièrement quadratique sur $B(x_0, \Delta)$*
   - *soit trouver un modèle $\tilde{m} \in \mathcal{M}$ qui est entièrement quadratique.*

Cette propriété plus forte permet d'assurer un comportement du modèle identique à un modèle construit avec des informations sur les dérivées. Elle donne ainsi la possibilité d'apporter des preuves de convergences de certains algorithmes vers un point critique d'ordre 2.

2.2.6.2 Lien avec la Λ-unisolvance

Il est possible de montrer qu'il existe des bornes similaires à celles données dans la définition 6 sur les erreurs entre des modèles construits avec un ensemble Λ-unisolvant et la fonction interpolée [27]. On travaille sous l'hypothèse suivante :

Hypothèse 1. *On suppose que la fonction objectif F est deux fois continûment dérivable avec une Hessienne continue et Lipschitzienne sur un domaine ouvert S contenant le point x_0 et la région de confiance $B(x_0, \Delta)$.*

Le premier résultat dans le cas de modèles déterminés est le suivant :

Proposition 3. *Sous l'hypothèse 1, on suppose qu'on dispose d'un point x_0 et d'un ensemble d'interpolation $Y = \{y_0, ..., y_{L_p}\}$ entièrement déterminé et Λ-unisolvant sur $B(x_0, \Delta)$ pour un certain Λ. Alors, il existe des constantes k_{eh}, k_{eg} et k_{ef} ne dépendant que de Λ, L_p et de la constante de Lipschitz de $\nabla^2 F$ telles que*

- *l'erreur entre la Hessienne du modèle et celle de F vérifie*

$$\forall x \in B(x_0, \Delta),\ \left\|\nabla^2 F(x) - \nabla^2 m(x)\right\| \leq k_{eh}\Delta,$$

- *l'erreur entre le gradient du modèle et celui de F vérifie*

$$\forall x \in B(x_0, \Delta),\ \|\nabla F(x) - \nabla m(x)\| \leq k_{eg}\Delta^2,$$

- *l'erreur entre le modèle et F vérifie*

$$\forall x \in B(x_0, \Delta),\ \|F(x) - m(x)\| \leq k_{ef}\Delta^3.$$

Dans le cas de modèles sous-déterminés, on ne peut que vérifier l'entière linéarité :

Proposition 4. *Sous l'hypothèse 1, on suppose qu'on dispose d'un point x_0 et d'un ensemble d'interpolation $Y = \{y_0, ..., y_q\}$ sous-déterminé et Λ-unisolvant sur $B(x_0, \Delta)$ pour un certain Λ. Alors, il existe des constantes k_{eg} et k_{ef} ne dépendant que de Λ, q et de la constante de Lipschitz de $\nabla^2 F$ telles que*

— *l'erreur entre le gradient du modèle et celui de F vérifie*

$$\forall x \in B(x_0, \Delta),\ \|\nabla F(x) - \nabla m(x)\| \leq k_{eg}\Delta,$$

— *l'erreur entre le modèle et F vérifie*

$$\forall x \in B(x_0, \Delta),\ \|F(x) - m(x)\| \leq k_{ef}\Delta^2.$$

Il est aussi possible de montrer les bornes suivantes :

Proposition 5. *Étant données une boule $B(x_0, \Delta)$, un ensemble d'interpolation Y Λ-unisolvant sur $B(x_0, \Delta)$ et sa base de polynômes de Lagrange associée $\{l_1, \ldots, l_q\}$, il existe des constantes $k_f > 0$ et $k_g > 0$ telles que, pour un modèle d'interpolation quadratique m et un point quelconque $y \in B(x_0, \Delta)$:*

$$\begin{cases} \|F(y) - m(y)\| \leq k_f \sum_{j=1}^{q} \|y_j - y\|^2 |l_j(y)| \\ \|\nabla F(y) - \nabla m(y)\| \leq k_g \Lambda \Delta. \end{cases}$$

La Λ-unisolvance d'un modèle permet donc concrètement d'estimer la qualité d'un modèle, ce qui s'avère très utile dans le cadre d'un algorithme d'optimisation.

2.2.6.3 Amélioration de la géométrie des ensembles d'interpolation

On a besoin à ce stade d'un algorithme capable, pour un ensemble d'interpolation Y donné, de :

— déterminer si Y est Λ-unisolvant ou non sur une boule B,

— modifier Y de façon à ce qu'il soit Λ-unisolvant sur B.

L'algorithme 5 donne une façon de modifier Y itérativement pour le rendre unisolvant en utilisant les polynômes de Lagrange.
On peut montrer le résultat suivant pour l'algorithme 5 [27] :

Lemme 1. *L'algorithme 5 se termine en un nombre fini d'itérations avec un ensemble Y Λ-unisolvant sur B.*

On dispose ainsi d'un outil permettant d'assurer en temps fini la Λ-unisolvance d'un ensemble d'interpolation donné. On peut donc à présent construire un algorithme beaucoup plus robuste que l'algorithme 4.

2.2.7 Un algorithme classique : NEWUOA

Étant donné un modèle quadratique, les propriétés des sections précédentes nous apportent un moyen de déterminer s'il est entièrement linéaire et de l'améliorer itérativement s'il ne l'est pas. Il est donc possible de remédier aux défauts de l'algorithme simpliste numéro 4. Dans cet algorithme, la réduction de la taille de la région de confiance ne garantissait pas une amélioration de la qualité du modèle. Cependant, ce problème disparait si on ajoute la possibilité de garantir

Algorithme 5 Un algorithme d'amélioration de la Λ-unisolvance

1. **Initialisation**
   — Générer un ensemble d'interpolation $Y = \{y_1, ..., y_q\}$.
   — Choisir une boule fermée $B \subset \mathbb{R}^p$ et une constante $\Lambda > 1$.
   — Construire les polynômes de Lagrange de norme de Frobenius minimale associés à Y, notés $\{l_1, ..., l_q\}$.
   — Fixer $k = 0$.
2. **Itération** k
   — Estimer
   $$\Lambda_k = \max_{1 \leq i \leq q} \max_{x \in B} l_i(x).$$
   — Si $\Lambda_k \leq \Lambda$, Y est Λ-unisolvant : sortir de l'algorithme.
   — Si $\Lambda_k > \Lambda$, soit i un indice tel que
   $$\max_{x \in B} |l_i(x)| > \Lambda$$
   et x_k un point de B tel que $|l_i(x_k)| > \Lambda$.
   Mettre à jour Y en remplaçant le point y_i par x_k.
   — Mettre à jour les polynômes de Lagrange.

l'entière linéarité des modèles. En effet, comme dans le cas des modèles calculés avec les dérivées des fonctions objectifs, il ne sera pas possible pour la région de confiance de diminuer indéfiniment hors des points critiques de F.

On peut noter que dans une optique d'optimisation, il n'est pas nécessaire de garantir la Λ-unisolvance du modèle à toutes les itérations de l'algorithme. Si à une certaine itération k le modèle a bien prédit la diminution de la fonction objectif, on pourra accepter le point d'essai x_k^+ sans vérifier si le modèle est proche de la fonction objectif sur toute la région de confiance. On introduit donc la quantité $\rho_k = \frac{F(x_k) - F(x_k^+)}{m_k(x_k) - m_k(x_k^+)}$ qui nous servira de critère pour déterminer si une itération est fructueuse ou non. On décrit dans l'algorithme 6 un algorithme classique très proche de l'algorithme NEWUOA de Powell [84, 86], mais aussi de l'algorithme SQA développé à IFPEN ([63]).

Algorithme 6 Algorithme NEWUOA ou SQA (variante IFPEN)

1. **Initialisation**
   - Choisir un point initial x_0 et un rayon de région de confiance Δ_0.
   - Choisir un ensemble d'interpolation Y_0 contenant $t \in [\![p+2, L_p]\!]$ points, dont x_0, et évaluer F sur tous les points de Y_0.
   - Construire le modèle d'interpolation de F initial, noté m_0.
   - Choisir les constantes $\epsilon > 0$, $\epsilon_0 > 0$, $q = 0$, $0 < \nu < 1$, $\gamma_{inc} \geq 1$, $0 < \gamma_{low} < 1$, $0 < \mu < 1$, $0 < \theta < 1$ et $\Lambda > 1$.
2. **Étape critique**
   (a) Fixer $\widehat{m_q} = m_k$.
   (b) Si $\|\nabla \widehat{m_q}(x_k)\| < \epsilon$, arrêter l'algorithme et retourner x_k.
   (c) Si $\|\nabla \widehat{m_q}(x_k)\| < \epsilon_q$, fixer $\epsilon_{q+1} = \mu \|\nabla \widehat{m_q}(x_k)\|$, construire un nouveau modèle Λ-unisolvant $\widehat{m_{q+1}}$ dans $B(x_k, \epsilon_{q+1})$ et incrémenter q.
   (d) Si q a été incrémenté, choisir $m_k = \widehat{m_q}$, $\Delta_k = \theta \|\nabla m_k(x_k)\|$ et recommencer l'étape du test critique à partir du pas 2(b).
3. **Iteration k**
   Calculer le point d'essai x_k^+ tel que :
   $$m_k(x_k^+) = \min_{x \in B(x_k, \Delta_k)} m_k(x)$$
   Évaluer $F(x_k^+)$, $\rho_k = \frac{F(x_k) - F(x_k^+)}{m_k(x_k) - m_k(x_k^+)}$, les polynômes de Lagrange $\{l_{k,1}, ..., l_{k,t}\}$ associés à Y_k et $y_{k,max} = arg\max_{y_{k,j} \in Y_k} \left\|y_{k,j} - x_k^+\right\|^2 \left|l_{k,j}(x_k^+)\right|$.
   (a) **Itération fructueuse :** Si $\rho_k \geq \nu$ ou si le modèle m_k est entièrement linéaire (Y_k est Λ-unisolvant) et $\rho_k \geq 0$, choisir $x_{k+1} = x_k^+$ et remplacer $y_{k,max}$ par x_k^+ dans Y_{k+1}.
   (b) **Itération infructueuse :** Si $\rho_k < \nu$, choisir $x_{k+1} = x_k$ et tenter de remplacer un point de Y_k en suivant une itération de l'algorithme 5. Le choix des nouveaux points se fait pour les variantes de Powell à l'intérieur d'une région de confiance de rayon plus petit que Δ_k. Si aucun point de Y_k n'a pu être remplacé, alors Y_k est Λ-unisolvant et on peut garantir l'entière linéarité de m_k.
   (c) **Gestion de la région de confiance :**
   - Si $\rho_k \geq \nu$, choisir $\Delta_{k+1} = \gamma_{inc}\Delta_k$.
   - Si $\rho_k < \nu$ et m_k est entièrement linéaire, choisir $\Delta_{k+1} = \gamma_{low}\Delta_k$.
   - Choisir $\Delta_{k+1} = \Delta_k$ dans les autres cas.
4. **Mise à jour du modèle**
   Calculer le nouveau modèle d'interpolation m_{k+1} centré autour de x_{k+1} à partir des points de Y_{k+1}, puis retourner à l'étape 2.

2.2.8 Auto-correction de la géométrie des ensembles d'interpolation

Comme observé au début de la section 2.2, la figure 2.1 issue de [98] est un exemple qui montre que les étapes d'amélioration de la géométrie de l'ensemble d'interpolation sont nécessaires pour assurer la convergence d'un algorithme à région de confiance vers un point critique de F. Cependant, Fasano et al. ont montré dans [39] que de très bons résultats numériques pouvaient être obtenus en les supprimant. Scheinberg et Toint [98] ont donc développé en 2010

un algorithme basé sur des modèles d'interpolation qui, sans ignorer totalement la qualité des modèles, réduit considérablement le nombre d'évaluations de la fonction objectif réalisé dans le seul but d'améliorer le modèle. L'idée exploitée pour parvenir à ce résultat est simplement d'étudier l'effet de l'addition du point d'essai x_k^+ à l'ensemble d'interpolation à chaque itération de l'algorithme. Il s'avère que pour la plupart des itérations, le point d'essai améliore la géométrie de l'ensemble d'interpolation et retire donc le besoin d'évaluer la fonction objectif en davantage de points.

Pour pouvoir décrire l'algorithme de Scheinberg et Toint [98], on se place dans un contexte d'optimisation sans dérivées avec des modèles d'interpolation du type de NEWUOA (algorithme 6). A l'itération k, le modèle est noté m_k, l'ensemble d'interpolation Y_k, le point d'essai x_k^+, le rayon de la région de confiance Δ_k et les polynômes de Lagrange associés à Y_k $\{l_{k,1}, ..., l_{k,q}\}$. On introduit pour une itération k et deux constantes $\Lambda > 1$ et $\beta > 1$ données :

— l'ensemble F_k des points d'interpolation éloignés de la région de confiance :

$$F_k = \{y_{k,j} \in Y_k \text{ tels que } \|y_{k,j} - x_k\| > \beta\Delta_k \text{ et } l_{k,j}(x_k^+) \neq 0\}, \tag{2.16}$$

— l'ensemble C_k des points proches de la région de confiance qui permettent une amélioration de la géométrie de l'ensemble d'interpolation s'ils sont remplacés par le point d'essai x_k^+

$$C_k = \{y_{k,j} \in Y_k \backslash \{x_k\} \text{ tels que } \|y_{k,j} - x_k\| \leq \beta\Delta_k \text{ et } l_{k,j}(x_k^+) \geq \Lambda\}. \tag{2.17}$$

Le résultat fondamental sur lequel Scheinberg et Toint se basent pour montrer la convergence de leur algorithme vers un point critique du premier ordre est le suivant.

Lemme 2. *Si la fonction objectif F est continûment dérivable et que son gradient est continu et Lipschitzien alors, pour tout $\Lambda > 1$, il existe une constante k_Λ telle que si l'itération k est infructueuse et que :*

$$\begin{cases} F_k = \emptyset, \\ \Delta_k < k_\Lambda \|\nabla m_k(x_k)\|, \end{cases}$$

alors :

$$C_k \neq \emptyset. \tag{2.18}$$

Ce résultat stipule que si tous les points de l'ensemble d'interpolation sont suffisamment proches de la région de confiance ($F_k = \emptyset$) et que le rayon de la région de confiance est suffisamment petit ($\Delta_k < k_\Lambda \|\nabla m_k(x_k)\|$), alors il existe un point dans Y_k dont le remplacement par x_k^+ améliorera la qualité du modèle ($C_k \neq \emptyset$). L'algorithme 7 exploite cette propriété et utilise le point d'essai de chaque itération pour tenter d'améliorer la géométrie de l'ensemble d'interpolation du modèle.

Algorithme 7 Un algorithme sans dérivées utilisant une propriété d'auto-correction des ensembles d'interpolation

1. **Initialisation**
   — Choisir un point initial x_0 et un rayon de région de confiance Δ_0.
   — Choisir un ensemble d'interpolation initial Y_0 contenant x_0 et évaluer F sur tous les points de Y_0.
   — Construire le modèle initial m_0 interpolant F aux points de Y_0.
   — Choisir les constantes $\epsilon > 0$, $\epsilon_0 > 0$, $0 < \nu < 1$, $\gamma_{inc} \geq 1$, $0 < \gamma_{low} < 1$, $\Lambda > 1$, $0 < \mu < 1$, $\theta > 0$ et $\beta \geq 1$.
   — choisir $v_0 \neq x_0$, fixer $q = 0$ et $k = 0$.
2. **Test critique**
   (a) Définir $\widehat{m_q} = m_k$.
   (b) Si $\|\nabla \widehat{m_q}(x_k)\| < \epsilon$, arrêter l'algorithme et retourner x_k.
   (c) Si $\|\nabla \widehat{m_q}(x_k)\| < \epsilon_q$, fixer $\epsilon_{q+1} = \mu \|\nabla \widehat{m_q}(x_k)\|$, construire un nouveau modèle $\widehat{m_{q+1}}$ Λ-unisolvant dans $B(x_k, \epsilon_{q+1})$ et incrémenter q.
   (d) Si q a été incrémenté, choisir $m_k = \widehat{m_q}$, $v_q = x_k$, $\Delta_k = \theta \|\nabla m_k(x_k)\|$ et recommencer l'étape du test critique à partir du pas 2(b).
3. **Itération k**
   Calculer le point d'essai x_k^+ tel que :

$$m_k(x_k^+) = \min_{x \in B(x_k, \Delta_k)} m_k(x).$$

   Évaluer $F(x_k^+)$ et $\rho_k = \frac{F(x_k) - F(x_k^+)}{m_k(x_k) - m_k(x_k^+)}$.
   (a) **Itération fructueuse :** Si $\rho_k \geq \nu$, définir $x_{k+1} = x_k^+$, $\Delta_{k+1} = \gamma_{inc} \Delta_k$ et $Y_{k+1} = Y_k \backslash \{y_{k,r}\} \cup \{x_k^+\}$ avec $y_{k,r}$ tel que :

$$y_{k,r} = arg \max_{y_{k,j} \in Y_k} \left\| y_{k,j} - x_k^+ \right\|^2 \left| l_{k,j}(x_k^+) \right|.$$

   (b) **Remplacer un point d'interpolation éloigné :** Si $\rho_k < \nu$, et que, soit $x_k \neq v_q$, soit $\Delta_k \leq \epsilon_q$ et l'ensemble $F_k \neq \emptyset$, alors choisir $x_{k+1} = x_k$, $\Delta_{k+1} = \Delta_k$ et $Y_{k+1} = Y_k \backslash \{y_{k,r}\} \cup \{x_k^+\}$ où $y_{k,r}$ est un point de F_k, par exemple :

$$y_{k,r} = arg \max_{y_{k,j} \in F_k} \left\| y_{k,j} - x_k^+ \right\|^2 \left| l_{k,j}(x_k^+) \right|$$

   (c) **Remplacer un point d'interpolation proche :** Si $\rho_k < \nu$, et que, soit $x_k \neq v_q$, soit $\Delta_k \leq \epsilon_q$, l'ensemble $F_k = \emptyset$ et l'ensemble $C_k \neq \emptyset$, alors choisir $x_{k+1} = x_k$, $\Delta_{k+1} = \Delta_k$ et $Y_{k+1} = Y_k \backslash \{y_{k,r}\} \cup \{x_k^+\}$ où $y_{k,r}$ est un point de C_k, par exemple :

$$y_{k,r} = arg \max_{y_{k,j} \in C_k} \left\| y_{k,j} - x_k^+ \right\|^2 \left| l_{k,j}(x_k^+) \right|$$

   (d) **Réduire la région de confiance :** Si $\rho_k < \nu$, et que, soit $x_k = v_q$ et $\Delta_k > \epsilon_q$, soit $F_k \cup C_k = \emptyset$, alors choisir $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_{low} \Delta_k$ et $Y_{k+1} = Y_k$.
4. **Mise à jour du modèle :** Calculer le nouveau modèle d'interpolation m_{k+1} centré autour de x_{k+1} à partir des points de Y_{k+1}. Incrémenter k et retourner à l'étape 2.

L'idée exploitée par l'algorithme 7 est de faire en sorte de se mettre le plus souvent possible dans la situation des hypothèses du lemme 2. Les étapes 3(b), 3(c) et 3(d) décrivent les différents cas qui se présentent lorsqu'une itération infructueuse survient :

(i) s'il existe des points d'interpolation éloignés de la région de confiance, on en remplace un par le point d'essai,

(ii) s'il n'y a pas de point éloigné mais que C_k est non vide, alors on peut améliorer la géométrie de Y_k en remplaçant un point de C_k par x_k^+,

(iii) si aucun des deux cas précédents ne survient, on peut réduire la région de confiance sans difficulté puisque :
   — soit le modèle n'est pas entièrement linéaire et on se rapproche des conditions du lemme 2, lequel garantira que C_k sera non vide lors d'une prochaine itération,
   — soit le modèle est entièrement linéaire et la réduction de la région de confiance diminuera l'écart entre le modèle et la fonction objectif.

On peut donc voir que, comme pour l'algorithme NEWUOA (algorithme 6), il n'est pas possible pour le rayon de la région de confiance de diminuer indéfiniment hors des points critiques de F.

2.3 Une nouvelle méthode de type région de confiance pour les fonctions partiellement séparables : l'algorithme DFO-PSOF

On présente dans cette section une méthode d'optimisation sans dérivées développée durant la thèse de façon à être adaptée aux fonctions partiellement séparables. On rappelle qu'une fonction est dite partiellement séparable si elle peut s'écrire sous la forme :

$$\begin{gathered} F(x_1, \ldots, x_p) = \sum_{i=1}^{n} f_i(x_{i,1}, \ldots, x_{i,p_i}) \\ \text{avec } \forall i \in \{1, ..., n\}, p_i < p \end{gathered} \tag{2.19}$$

L'objectif ici est d'exploiter au maximum cette propriété de façon à réduire le nombre d'évaluations de la fonction objectif pour atteindre un optimum local. Au lieu de créer un modèle quadratique pour la fonction objectif complète F comme dans l'algorithme 6 NEWUOA, l'idée principale consiste à créer un modèle pour chaque sous-fonction f_i : à l'itération k de l'algorithme, le modèle m_k de F s'écrit alors

$$m_k(x_1, ..., x_p) = \sum_{i=1}^{n} m_k^i(x_{i,1}, \ldots, x_{i,p_i}) \tag{2.20}$$

On espère ainsi avoir besoin de moins de points pour construire le modèle tout en réduisant l'erreur commise entre le modèle et la fonction objectif.

La manière la plus simple d'obtenir un algorithme exploitant la propriété de la fonction objectif serait de choisir un algorithme classique du type de NEWUOA et d'adapter chaque étape pour gérer un modèle par sous-fonction. A une itération k, on pourrait par exemple :

(a) Si l'itération est fructueuse, remplacer un point dans chaque sous-ensemble d'interpolation Y_k^i par x_k^+.

(b) Si l'itération n'est pas fructueuse et que tous les ensembles d'interpolation ne sont pas entièrement linéaires, remplacer un point de chaque ensemble d'interpolation non entièrement linéaire en suivant une itération de l'algorithme 5.

Plusieurs problèmes apparaissent avec une telle adaptation de NEWUOA. D'abord, l'initialisation du premier modèle m_0 peut potentiellement requérir plus d'évaluations de la fonction objectif si la séparabilité partielle de F est prise en compte. En effet, si dans NEWUOA on choisit d'initialiser m_0 avec $t(p)$ points (par exemple $2p+1$ points), on devra, pour garder la même précision dans l'algorithme adapté, initialiser chaque sous-modèle avec $t(p_i)$ points ($2p_i+1$ points dans l'exemple). Une telle initialisation nécessite l'évaluation de $\sum_{i=1}^{n} t(p_i)$ points. Ce nombre est dépendant de la structure de F et peut très largement excéder $t(p)$, ce qui n'est pas acceptable dans une optique de réduction du coût de l'optimisation.

De plus, afin de maintenir la qualité de tous les sous-modèles au cours de l'optimisation, l'algorithme adapté pourrait requérir à chaque itération l'évaluation de F en plusieurs points. Le coût de telles évaluations peut très vite devenir élevé sans garantir une accélération de la convergence de l'algorithme. Colson et Toint [23] on proposé un moyen de réduire le nombre de points à évaluer dans une étape similaire mais leur algorithme nécessite toujours l'évaluation potentielle de la fonction objectif en de multiples points à chaque itération.

Nous proposons dans la suite de la section une méthode pour initialiser le modèle m_0 de façon plus efficace ainsi qu'une généralisation de la propriété d'auto-correction de la géométrie aux fonctions partiellement séparables.

2.3.1 Initialisation de l'algorithme DFO-PSOF

Pour faire en sorte que le modèle initial m_0 soit suffisamment précis dans un algorithme comme NEWUOA, il est courant de travailler avec des ensembles d'interpolation contenant $2p+1$ points. Ce choix pour le nombre de points à évaluer initialement offre un bon compromis entre coût et précision du modèle puisque qu'il permet d'utiliser de l'information sur la courbure de la fonction objectif tout en restant de l'ordre de p. Dans cette partie, on initialisera donc chacun des sous-modèles avec $2p_i + 1$ points.

(i) Un exemple simple (cas entièrement séparable)
On commence par considérer un exemple très simple dans lequel F ne dépend que de 2 paramètres x_1 et x_2, et peut s'écrire sous la forme :

$$F(x_1, x_2) = f_1(x_1) + f_2(x_2) \tag{2.21}$$

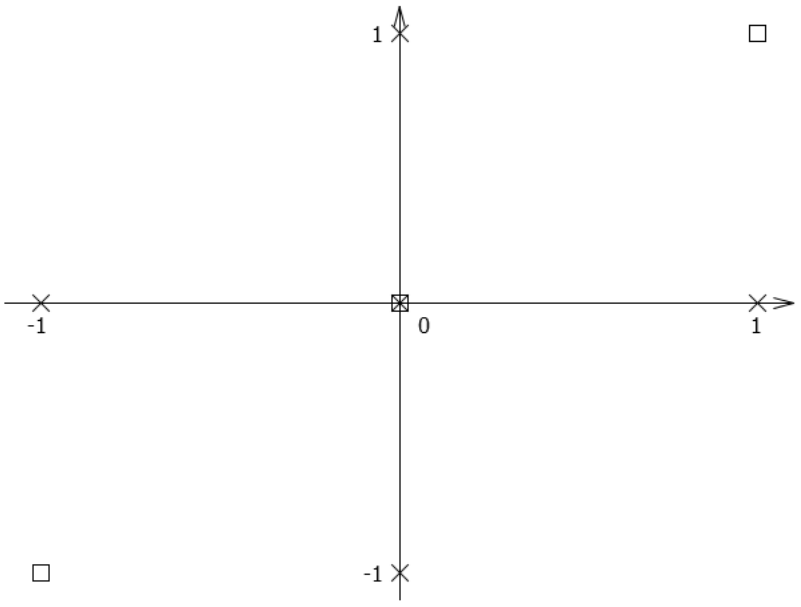


FIGURE 2.3 – Les croix sont les points à évaluer pour construire un modèle quadratique avec $2p+1$ points sans exploiter la séparabilité partielle tandis que les carrés sont ceux utilisés pour construire les modèles de f_1 et f_2.

On veut construire un modèle d'interpolation m de F autour de $x_0 = (0,0)$ dans la boule $B(x_0, 1)$. Pour construire directement m avec $2p + 1 = 5$ points, on choisit généralement les points représentés par les croix sur la figure 2.3. Avec les mêmes points, on peut construire les modèles initiaux m_1 et m_2 de f_1 et f_2 : l'initialisation de m_1 se fait avec les 3 points $(-1, 0)$, $(0, 0)$ et $(1, 0)$ tandis que celle de m_2 se fait à l'aide des points $(0, -1)$, $(0, 0)$ and $(0, 1)$. Cependant, cette façon de faire ne prend pas en compte le fait que f_1 est indépendant de x_2 et que f_2 est indépendant de x_1. En fait, sans perdre d'information, on peut réduire les points d'interpolation aux seuls 3 points représentés par des carrés sur la figure 2.3 : le point $(1, 1)$ peut remplacer à la fois les points $(0, 1)$ et $(1, 0)$ tandis que $(-1, -1)$ peut remplacer $(0, -1)$ et $(-1, 0)$ pour la construction des modèles m_1 and m_2.

C'est cette façon de faire que l'on va généraliser à l'ensemble des fonctions partiellement séparables dans la suite. Pour cela, on commence par chercher à déterminer les indépendances entre les différentes variables à l'aide d'un graphe coloré. Une fois ces indépendances identifiées, on perturbe les points d'interpolation simultanément dans toutes les directions mutuellement indépendantes.

(ii) Coloration des variables

Afin de généraliser l'exemple précédent, on commence par modéliser les dépendances des sous-fonctions de F par un graphe. Ce graphe suit les règles suivantes [58] :

— chaque paramètre x_i est représenté par un nœud i,

— s'il existe une sous-fonction f_k de F qui dépend à la fois des variables x_i et x_j , une arête est créée entre les nœuds i et j du graphe.

Exemple : on veut construire le graphe de dépendance de la fonction

$$F(x_1, x_2, x_3, x_4) = f_1(x_1, x_2) + f_2(x_2, x_3) + f_3(x_3, x_4) + f_4(x_4, x_1) + f_5(x_2, x_4) \tag{2.22}$$

Le graphe qui lui est associé est schématisé dans la figure 2.4.

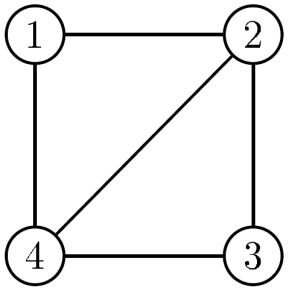


FIGURE 2.4 – Graphe de dépendance de la fonction F définie en (2.22).

Afin d'identifier les différentes indépendances entre les variables, on colore le graphe de dépendance [28, 22]. On cherche donc à attribuer une couleur à chaque nœud du graphe de dépendance de la fonction objectif en suivant les deux règles suivantes :

— on colore avec le moins de couleurs différentes possibles,

— deux nœuds reliés par un arête ne peuvent pas avoir la même couleur.

Avec un graphe de dépendance coloré, il devient facile d'identifier les variables indépendantes entre elles. En effet, deux nœuds i et j du graphe auront la même couleur seulement s'il n'existe pas de sous-fonction de F qui dépende à la fois de x_i et de x_j. On montre en figure 2.5 la coloration du graphe de dépendance de F.

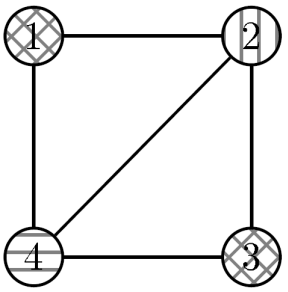


FIGURE 2.5 – Graphe de dépendance coloré de la fonction F (2.22).

Pour la fonction F, les paramètres x_1 et x_3 sont indépendants (il n'existe pas de sous-fonction de F qui dépende à la fois de x_1 et de x_3), on peut donc attribuer la même couleur aux nœuds 1 et 3.

(iii) Initialisation des sous-modèles
Pour que les sous-modèles m_0^i aient une certaine précision, on veut toujours qu'ils soient chacun construits avec $2p_i + 1$ points d'interpolation. Afin d'y parvenir, on exploite les indépendances des différents paramètres. On dit que deux paramètres x_1 et x_2 sont indépendants si les sous-fonctions qui dépendent d'un des paramètres ne dépendent pas de l'autre. Par conséquent, la valeur du paramètre x_2 des points d'interpolation servant à construire les modèles dépendants de x_1 n'a pas d'influence. La coloration du graphe de dépendance de la fonction objectif nous permet d'identifier ces indépendances.

Soit c le nombre de couleurs différentes utilisées pour colorer le graphe de dépendance de F. On choisit seulement $2c + 1$ points $Y_0 = \{y_0, ..., y_{2c}\}$ pour initialiser tous les sous-modèles sur $B(x_0, \Delta_0)$:

- $y_0 = x_0$
- $\forall i \in \{1, ..., c\}$, on appelle c_i la $i^{\text{ème}}$ couleur et I_i l'ensemble des indices des points colorés par la couleur c_i
- Les points choisis sont alors

$$\forall i \in \{0, ..., c-1\} \begin{cases} y_{i+1} = y_0 + \Delta_0 \sum_{j \in I_i} e_j \\ y_{i+c+1} = y_0 - \Delta_0 \sum_{j \in I_i} e_j \end{cases} \tag{2.23}$$

Le nombre de nœuds du graphe de dépendances étant le nombre de paramètres dont dépend F, on a toujours $c \leq p$. On est donc parvenu à réduire systématiquement le nombre de points à évaluer pour la construction du modèle initial m_0 sans influer sur sa précision. Ce nombre d'évaluation dépend en fait uniquement de la séparation de F. Un exemple simple permettant de comprendre le gain potentiel apporté par cette façon de faire est celui des fonctions entièrement séparables. Si F peut s'écrire sous la forme

$$F(x_1, ..., x_p) = f_1(x_1) + \cdots + f_p(x_p) \tag{2.24}$$

dans ce cas, tous les paramètres sont indépendants et sont donc colorés de la même couleur. Cette propriété implique que, quelque soit le nombre de paramètres p dont dépend F, on pourra toujours construire le premier modèle m_0 à l'aide de seulement 3 points. En fait, moins il y a d'interactions entre les variables, moins le nombre de points requis pour construire m_0 sera élevé.

2.3.2 Une propriété d'auto-correction de la géométrie des ensembles d'interpolation

On a vu en début de section 2.3 que le maintien de la qualité de tous les sous-modèles dans un algorithme à région de confiance pouvait avoir un coût beaucoup trop important pour pouvoir être utilisé dans la pratique. Afin de réduire ce coût, on généralise le lemme 2 de la section précédente qui stipule qu'un algorithme basé sur des modèles d'interpolation corrige généralement de lui-même la géométrie de l'ensemble d'interpolation.

Avant de donner quelques notations et définitions, on remarque que chaque sous-modèle m_k^i ne dépend que de vecteurs de dimension p_i appartenant au sous-espace $Vect\{e_{i,1}, \ldots, e_{i,p_i}\}$. Un sous-modèle m_k^i est donc construit à partir de l'évaluation de f_i aux projections orthogonales

des points d'interpolation sur le sous-espace associé. Afin de ne pas alourdir les notations, on garde ces projections implicites dans toute la suite.

On introduit à présent la notion de sous-modèle "dominant" :

Définition 8. *Dans un algorithme à région de confiance adapté aux fonctions partiellement séparables, à l'itération k, le sous-modèle m_k^j est appelé sous-modèle dominant si*

$$j = arg \max_{i \in \{1,\dots,n\}} \big(m_k^i(x_k) - m_k^i(x_k^+)\big). \tag{2.25}$$

À l'itération k, le sous-modèle dominant est simplement celui qui a le plus contribué à la diminution du modèle global m_k.

De façon similaire à la section 2.2.8, on introduit les ensembles F_k^i et C_k^i :

Définition 9. *On se place à l'itération k d'un algorithme à région de confiance adapté aux fonctions partiellement séparables. Pour tout indice i, si Y_k^i est l'ensemble d'interpolation du sous-modèle m_k^i, $\beta \geq 1$ et $\Lambda > 1$ des réels et $\{l_{k,h}\}_h$ les polynômes de Lagrange associés à Y_k^i, alors les ensembles F_k^i et C_k^i sont respectivement définis par :*

— *l'ensemble des points d'interpolation de m_k^i éloignés de la région de confiance*

$$F_k^i = \{y_{k,h} \in Y_k^i \text{ tels que } \|y_{k,h} - x_k\| > \beta\Delta_k \text{ et } l_{k,h}(x_k^+) \neq 0\}, \tag{2.26}$$

— *l'ensemble des points d'interpolation de m_k^i qui permettent d'améliorer le modèle s'ils sont remplacés par le point d'essai x_k^+*

$$C_k^i = \{y_{k,h} \in Y_k^i \backslash \{x_k\} \text{ tels que } \|y_{k,h} - x_k\| \leq \beta\Delta_k \text{ et } l_{k,h}(x_k^+) \geq \Lambda\}. \tag{2.27}$$

On définit ensuite les quantités ρ_k et ρ_k^i pour chaque sous-modèle m_k^i qui servent de critère pour décider si la qualité des modèles est suffisante :

Définition 10. *Dans un algorithme à région de confiance adapté aux fonctions partiellement séparables, si x_k^+ est le point d'essai proposé à l'itération k et $i \leq n$ est l'indice d'une sous-fonction f_i, on définit :*

$$\rho_k = \frac{F(x_k) - F(x_k^+)}{m_k(x_k) - m_k(x_k^+)} \tag{2.28}$$

et

$$\rho_k^i = \frac{f_i(x_k) - f_i(x_k^+)}{m_k^i(x_k) - m_k^i(x_k^+)} \tag{2.29}$$

Pour simplifier les notations dans l'algorithme, pour une itération k et un indice $i \in \{1, ..., n\}$ d'une sous-fonction f_i, on définit enfin p_k^i et L_p^i.

Définition 11. *Soit p_k^i le nombre de points d'interpolation utilisés pour construire le sous-modèle m_k^i et L_p^i le nombre de points d'interpolation nécessaire pour que le modèle m_k^i soit déterminé :*

$$\begin{cases} p_k^i = \left|Y_k^i\right| \\ L_p^i = \frac{(p_i+1)(p_i+2)}{2}. \end{cases} \tag{2.30}$$

Avec les notations précédentes, on pourra énoncer un résultat d'auto-correction de la géométrie des ensembles d'interpolation. Ce résultat (énoncé et démontré en section 2.4) stipule que, sous certaines conditions sur la région de confiance, il est possible d'améliorer la Λ-unisolvance de l'ensemble d'interpolation du sous-modèle dominant en y remplaçant un point par le point d'essai x_k^+ proposé par l'algorithme à l'itération k. On montrera en section 2.4 que cette propriété est suffisante pour montrer la convergence de l'algorithme associé, présenté ci-dessous (algorithme 8).

2.3.3 Description de l'algorithme DFO-PSOF

Avec les propriétés et définitions précédentes, on peut à présent décrire le nouvel algorithme à région de confiance adapté aux fonctions partiellement séparables (algorithme 8), appelé DFO-PSOF ("Derivative Free Optimization for Partially Separable Functions").

Algorithme 8 Algorithme d'optimisation sans dérivées adapté aux fonctions partiellement séparables (DFO-PSOF).

---

1. **Initialisation**
   - Choisir un point initial x_0 et un rayon de région de confiance Δ_0.
   - Choisir des ensembles d'interpolation unisolvants Y_0^i suivant la méthode montrée en section 2.3.1 et évaluer F sur tout les points des ensembles Y_0^i.
   - Construire les modèles initiaux m_0^i interpolant les sous-fonctions f_i sur les ensembles Y_0^i.
   - Construire le modèle initial global $m_0 = \sum_{i=1}^{n} m_0^i$.
   - Choisir les constantes $q = 0$, $0 < \nu < 1$, $0 < \gamma < 1$, $0 < \mu \leq 1$, $\beta > 1$, $\epsilon_0 > 0$, $\epsilon > 0$, $\theta > 0$ et $\Lambda > 1$
   - Fixer k à 0 et choisir $v_0 \neq x_0$.
2. **Test critique**
   (a) Fixer $\widehat{m_q} = m_k$.
   (b) Si $\|\nabla \widehat{m_q}(x_k)\| < \epsilon$, arrêter l'algorithme et retourner x_k.
   (c) Si $\|\nabla \widehat{m_q}(x_k)\| < \epsilon_q$ fixer $\epsilon_{q+1} = \mu \|\nabla \widehat{m_q}(x_k)\|$, calculer un nouveau modèle Λ-unisolvant $\widehat{m_{q+1}}$ dans $B(x_k, \epsilon_{q+1})$ et incrémenter q.
   (d) Si q a été incrémenté, choisir $m_k = \widehat{m_q}$, $v_q = x_k$, $\Delta_k = \theta \|\nabla m_k(x_k)\|$ et recommencer l'étape critique à partir du pas 2(b).
3. **Itération k**
   (a) **Calculer le point d'essai :**
      - Calculer x_k^+ tel que $m_k(x_k^+) = \min_{x \in B(x_k, \Delta_k)} m_k(x)$.
      - Calculer pour tout $i \in \{1, \ldots, n\}$, $f_i(x_k^+)$, $f(x_k^+)$, ρ_k^i, et ρ_k.
   (b) **Suivre la procédure de gestion des sous-modèles (algorithme 9)**
   (c) **Gestion de la région de confiance :**
   Soit j l'indice du sous-modèle dominant m_k^j :
      - Si $\rho_k \geq \nu$, choisir $x_{k+1} = x_k^+$ et $\Delta_{k+1} \geq \Delta_k$.
      - Sinon, si $C_k^j \cup F_k^j = \emptyset$, choisir $x_{k+1} = x_k$ et $\Delta_{k+1} = \Delta_k$.
      - Sinon, si $\rho_k^j > \nu$ et au moins un point a été ajouté dans un sous-modèle dans l'étape 2 de la procédure de gestion des ensembles d'interpolation (algorithme 9), choisir $x_{k+1} = x_k$ et $\Delta_{k+1} = \Delta_k$.
      - Sinon si $\rho_k^j > \nu$ et aucun point n'a été ajouté à l'étape 2 de l'algorithme de la procédure de gestion des ensembles d'interpolation (algorithme 9), choisir $x_{k+1} = x_k$ et $\Delta_{k+1} = \gamma \Delta_k$.
      - Sinon si soit $x_k = v_q$ et $\Delta_k \geq \epsilon_q$ soit $C_k^j \cup F_k^j = \emptyset$, choisir $x_{k+1} = x_k$ et $\Delta_{k+1} = \gamma \Delta_k$.
   (d) **Incrémenter k, mettre à jour les modèles et retourner à l'étape 2**

Dans cet algorithme, on essaie d'exploiter au maximum un équivalent du lemme 2, en l'occur-

Algorithme 9 Gestion de l'ensemble d'interpolation des sous-modèles à l'itération k de l'algorithme 8.

1. **Pour tous les sous-modèles m_k^i, faire :**
   (a) Si m_k^i n'est pas entièrement déterminé ($p_k^i < L_p^i$),
   choisir $Y_{k+1}^i = Y_k^i \cup \{x_k^+\}$.
   (b) Sinon, si l'itération k est fructueuse ($\rho_k > \nu$),
   choisir $Y_{k+1}^i = Y_k^i \backslash \{y_{k,r}\} \cup \{x_k^+\}$ où $r = arg\max_h \left\|y_{k,h} - x_k^+\right\|^2 \left|l_{k,h}(x_k^+)\right|$.
   (c) Sinon, si $\rho_k^i \geq \nu$ et que soit $x_k \neq v_q$, soit $\Delta_k < \epsilon_q$,
   choisir $Y_{k+1}^i = Y_k^i \backslash \{y_{k,r}\} \cup \{x_k^+\}$ où $r = arg\max_h \left\|y_{k,h} - x_k^+\right\|^2 |l_{k,h}(x_k)|$.
   (d) Sinon, si soit $x_k \neq v_q$, soit $\Delta_k < \epsilon_q$ et que $F_k^i \neq \emptyset$,
   remplacer un point de F_k^i par x_k^+ dans Y_k^i.
   (e) Sinon, si soit $x_k \neq v_q$, soit $\Delta_k < \epsilon_q$ et que $C_k^i \neq \emptyset$
   remplacer un point de C_k^i par x_k^+ dans Y_k^i.
2. **Si : soit $x_k \neq v_q$, soit $\Delta_k < \epsilon_q$, et que $\rho_k < \nu$ et $\rho_k^j \geq \nu$, améliorer les sous-modèles non dominants :**
   Pour tous les sous-modèles tels que $\rho_k^i < \nu$, fixer $y_{k,r} = arg\max_{y_{k,h} \in Y_k^i} \left\|y_{k,h} - x_k^+\right\|^2 \left|l_{k,h}(x_k^+)\right|$. Si Y_k^i n'es pas Λ-unisolvant, remplacer $y_{k,r}$ en suivant une itération de l'algorithme 5.

rence le lemme 4 énoncé ultérieurement en section 2.4 concernant la propriété d'auto-correction de la géométrie des sous-ensembles d'interpolation. À chaque itération, soit on fait en sorte de se rapprocher des hypothèses du lemme 4, soit on garantit qu'au moins le sous-modèle dominant de l'itération est amélioré. L'algorithme ne requiert donc presque jamais l'évaluation de F en de nouveaux points uniquement pour assurer la qualité des modèles. Des points supplémentaires sont toutefois nécessaires dans le rare cas où l'itération globale est infructueuse mais le sous-modèle dominant est précis (étape 2 de l'algorithme permettant la gestion des ensembles d'interpolation, algorithme 9). Dans ce cas, on recommande d'améliorer tous les sous-modèles non-dominants qui ne sont pas assez prédictifs (tels que $\rho_k^i < \nu$). Il est cependant intéressant de noter que la convergence de l'algorithme peut toujours être prouvée avec l'amélioration d'un unique sous-modèle. Cela signifie que si un sous-modèle non dominant a été amélioré à l'étape 1 de l'algorithme 9, aucun point supplémentaire n'a besoin d'être évalué dans l'étape 2 du même algorithme.

Un autre point intéressant à signaler est le fait que, similairement à l'algorithme proposé par Tröltzsch dans [107], tant que les sous-modèles ne sont pas entièrement déterminés, l'algorithme DFO-PSOF continue d'ajouter les points d'essai dans leurs ensembles d'interpolation respectifs. L'une des spécificités du cas des fonctions partiellement séparables est en effet le fait qu'un sous-modèle est entièrement déterminé lorsqu'il est construit avec L_p^i points. Si F est correctement séparée (p_i est relativement petit devant p pour tout i), L_p^i est beaucoup plus petit que L_p, ce qui entraine une obtention beaucoup plus rapide de modèles entièrement déterminés. Dans ce cas, les modèles quadratiques de la fonction objectif seront précis avec beaucoup moins d'évaluations de la fonction objectif que dans un algorithme d'optimisation sans dérivées classique.

2.4 Preuve de convergence de l'algorithme DFO-PSOF

On se place dans le cadre des définitions de la section 2.3.2 et sous les hypothèses suivantes :

Hypothèse 2. *On suppose que :*
- *F est continûment dérivable et son gradient est continu et Lipschitzien de constante de Lipschitz $\frac{1}{2}L$,*
- *il existe une constante $k_{low} \in \mathbb{R}$ telle que $F(x) \geq k_{low}$ pour tout $x \in \mathbb{R}^p$,*
- *il existe une constante $k_H \geq L$ telle que pour toutes les itérations de l'algorithme, $1 + \|H_k\| \leq k_H$ où $H_k = \nabla^2 m_k(x_k)$.*

Le résultat de convergence de l'algorithme 8 vers un point critique d'ordre 1 est le suivant :

Théorème 1. *Sous les hypothèses précédentes, la suite $(x_k)_{k\in\mathbb{N}}$ construite dans l'algorithme 8 (DFO-PSOF) est telle que :*

$$\lim_{k\to+\infty} \inf \|\nabla F(x_k)\| = 0$$

Dans l'algorithme 8, le point d'essai est obtenu à l'itération k par minimisation du modèle m_k sur la région de confiance. On ne requiert en fait ici que la condition de Cauchy de descente, à savoir l'existence de $0 < k_c \leq 1$ tel que pour tout $k \in \mathbb{N}$:

$$m_k(x_k) - m_k(x_k^+) \geq k_c \|g_k\| \min\left(\frac{\|g_k\|}{1 + \|H_k\|}, \Delta_k\right) \tag{2.31}$$

où g_k est le gradient du modèle m_k en x_k et H_k la Hessienne.
Pour démontrer le théorème, on aura besoin du Lemme issu de [98] et rappelé ci-dessous :

Lemme 3. *Supposons que pour des réels $\{\alpha_i\}_{1\leq i\leq t}$ tels que*

$$\sigma_{abs} = \sum_{i=1}^{t} |\alpha_i| > 2\sum_{i=1}^{t} \alpha_i = \sigma > 0,$$

si on définit

$$i^* = arg \max_{i=1,\dots,t} |\alpha_i| \ \ et\ j^* = arg \max_{\substack{j=1,\dots,t \\ j\neq i^*}} |\alpha_j|$$

alors

$$|\alpha_{j^*}| \geq \frac{\sigma_{abs} - 2\sigma}{2t} \tag{2.32}$$

On montre maintenant le résultat central sur la nature auto-correctrice de l'algorithme 8 :

Lemme 4. *À l'itération k de l'algorithme 8, pour tout $\Lambda > 1$, il existe une constante k_Λ telle que :*

$$si \begin{cases} \text{- } j \text{ est l'indice du sous-modèle dominant} \\ \text{- } \Delta_k \leq k_\Lambda \|g_k\| \\ \text{- } \rho_k^j < \nu \\ \text{- } F_k^j = \emptyset \end{cases}$$

alors

$$C_k^j \neq \emptyset.$$

Démonstration. On montre le résultat pour la valeur particulière de k_Λ suivante :

$$k_\Lambda = \min\left(\frac{1}{k_H}, \frac{(1-\nu)\frac{k_c}{n}}{2k_f(\beta+1)^2(L_p^{max}\Lambda+1)}\right)$$

où $L_p^{max} = max_{i\in[\![1,\dots,n]\!]}L_p^i$ et k_f est la constante donnée dans la proposition 5. Sous les hypothèses et notations du lemme 4, on a

$$\rho_k^j = \frac{f_j(x_k) - f_j(x_k^+)}{m_k^j(x_k) - m_k^j(x_k^+)} < \nu.$$

Comme les sous-modèles interpolent les sous-fonctions f_i, on a aussi $\forall i \in \{1,...,n\}$, $f_i(x_k) = m_k^i(x_k)$. Cela implique

$$\begin{gathered} f_j(x_k^+) > (1-\nu)m_k^j(x_k) + \nu m_k^j(x_k^+), \\ \text{et donc } \left|f_j(x_k^+) - m_k^j(x_k^+)\right| > (1-\nu)\left|m_k^j(x_k) - m_k^j(x_k^+)\right|. \end{gathered}$$

De plus, la proposition 5 donne l'existence d'une constante $k_f > 0$ tel que

$$\left|f_j(x_k^+) - m_k^j(x_k^+)\right| \leq k_f \sum_{h=1}^{p_k^j} \left\|y_{k,h}^j - x_k^+\right\|^2 \left|l_{k,h}^j(x_k^+)\right|.$$

D'un autre côté, les hypothèses donnent $F_k^j = \emptyset$, donc

$$\forall h \in \{1,...,p_k^j\} \text{ tel que } l_{k,h}^j(x_k^+) \neq 0,\ \left\|y_{k,h}^j - x_k\right\| \leq \beta\Delta_k.$$

Ce qui donne, $\forall h \in \{1,...,p_k^j\}$ tel que $l_{k,h}^j(x_k^+) \neq 0$,

$$\left\|y_{k,h}^j - x_k^+\right\| \leq \left\|y_{k,h}^j - x_k\right\| + \left\|x_k^+ - x_k\right\| \leq (\beta+1)\Delta_k$$

et donc

$$(1-\nu)\left|m_k^j(x_k) - m_k^j(x_k^+)\right| < \left|f_j(x_k^+) - m_k^j(x_k^+)\right| \leq k_f(\beta+1)^2\Delta_k^2 \sum_{h=1}^{p_k^j}\left|l_{k,h}^j(x_k^+)\right|.$$

La condition de Cauchy donne quant à elle :

$$m_k(x_k) - m_k(x_k^+) \geq k_c \left\|g_k\right\| \min\left[\frac{\|g_k\|}{1+\|H_k\|}, \Delta_k\right].$$

Or, avec notre choix de k_Λ, on a par hypothèse :

$$\Delta_k \leq \frac{\|g_k\|}{k_H} \leq \frac{\|g_k\|}{1+\|H_k\|}$$

Donc, la condition de Cauchy peut se réécrire :

$$m_k(x_k) - m_k(x_k^+) \geq k_c \left\|g_k\right\| \Delta_k.$$

L'indice j correspondant au sous-modèle dominant, on a d'autre part

$$m_k^j(x_k) - m_k^j(x_k^+) \geq \frac{1}{n}(m_k(x_k) - m_k(x_k^+))$$

ce qui implique, avec la condition de Cauchy :

$$m_k^j(x_k) - m_k^j(x_k^+) \geq \frac{k_c}{n} \|g_k\| \Delta_k$$

et

$$\sum_{h=1}^{p_k^j} \left| l_{k,h}^j(x_k^+) \right| \geq \frac{(1-\nu)\frac{k_c}{n} \|g_k\|}{k_f(\beta+1)^2 \Delta_k},$$

et donc, avec l'hypothèse prise sur Δ_k,

$$\sum_{h=1}^{p_k^j} \left| l_{k,h}^j(x_k^+) \right| \geq 2(L_p^{max}\Lambda + 1) \geq 2(L_p^j\Lambda + 1) \geq 2(p_k^j\Lambda + 1).$$

Les $l_{k,h}^j$ étant des polynômes de Lagrange, on a aussi

$$\sum_{h=1}^{p_k^j} l_{k,h}^j(x_k^+) = 1.$$

Le lemme 3 rappelé précédemment garantit que si $m = arg\max_{h=1,\ldots,p_k^j} \left| l_{k,h}^j(x_k^+) \right|$,

$$\left| l_{k,r}^j(x_k^+) \right| \geq \Lambda \text{ pour } r = arg \max_{\substack{h=1,\ldots,p_k^j \\ h \neq m}} \left| l_{k,h}^j(x_k^+) \right|.$$

Cette dernière inégalité combinée au fait que $F_k^j = \emptyset$ prouve que $C_k^j \neq \emptyset$.

□

Le lemme suivant fait le lien entre la qualité des sous-modèles et la qualité du modèle global. Plus précisément, si tous les sous-modèles sont Λ-unisolvants et que la région de confiance est suffisamment petite, alors l'itération ne peut pas être infructueuse.

Lemme 5. *À l'itération k, si tous les ensembles d'interpolation Y_k^i sont Λ-unisolvants et*

$$\Delta_k < \min\left((1-\nu)\frac{k_c}{k_{ef}n}, \frac{1}{k_H} \right) \|g_k\|$$

où k_{ef} est la constante donnée par la proposition 4, alors $\rho_k > \nu$.

Démonstration. Par définition

$$\begin{aligned} \rho_k &= \frac{f(x_k) - f(x_k^+)}{m_k(x_k) - m_k(x_k^+)} = \frac{\sum_{i=1}^n \left(f_i(x_k) - f_i(x_k^+) \right)}{m_k(x_k) - m_k(x_k^+)} \\ &= \frac{\sum_{i=1}^n \left(f_i(x_k) - m_k^i(x_k) - f_i(x_k^+) + m_k^i(x_k^+) \right)}{m_k(x_k) - m_k(x_k^+)} + 1, \end{aligned}$$

La condition de Cauchy (2.31) donne comme précédemment :

$$m_k(x_k) - m_k(x_k^+) > k_c \|g_k\| \Delta_k,$$

tandis que les propriétés d'interpolation donnent :

$$f_i(x_k) = m_k^i(x_k),$$

et la Λ-unisolvance garantit l'entière linéarité d'après la proposition 4 :

$$\forall i \in \{1, \ldots, n\}, \left| f_i(x_k^+) - m_k^i(x_k^+) \right| \leq k_{ef} \Delta_k^2.$$

Les équations et inégalités précédentes impliquent alors que :

$$\begin{aligned} \left| \frac{\sum_{i=1}^n \left(f_i(x_k) - m_k^i(x_k) - f_i(x_k^+) + m_k^i(x_k^+) \right)}{m_k(x_k) - m_k(x_k^+)} \right| &< \frac{k_{ef} \Delta_k^2 n}{k_c \left\| g_k \right\| \Delta_k} \\ &< (1 - \nu) \end{aligned}$$

ce qui prouve donc le résultat annoncé, à savoir $\rho_k > \nu$. □

Avec ces premiers résultats, on peut à présent suivre les étapes classiques de preuve de convergence des méthodes à région de confiance. On commence par montrer que le rayon de la région de confiance Δ_k ne peut pas devenir arbitrairement petit hors d'un point critique.

Lemme 6. *Supposons qu'il existe $c > 0$ et $k_0 \geq 0$ tels que pour $k \geq k_0$:*

$$\|g_k\| \geq c,$$

alors il existe une constante $k_\Delta > 0$ telle que pour tout $k \geq k_0$

$$\Delta_k > k_\Delta.$$

Démonstration. Supposons que pour une itération $k \geq k_0$, $\Delta_k < \min\left((1-\nu)\frac{k_c}{k_{ef} n}, \frac{1}{k_H}, k_\Lambda, \mu \right) c$.

Soit k_q l'indice du modèle créé par la dernière étape de test critique, on a

$$\Delta_k < \mu c \leq \mu \left\| g_{k_q} \right\| = \epsilon_q,$$

donc $\Delta_k < \epsilon_q$.
Seuls cinq cas peuvent alors survenir :

1. $\rho_k \geq \nu$: dans ce cas, l'étape de gestion de la région de confiance (étape 3(c)) garantit que $\Delta_{k+1} \geq \Delta_k$.
2. $\rho_k < \nu$, $\rho_k^j \geq \nu$ et au moins un des sous-modèles n'est pas Λ-unisolvant : dans ce cas l'algorithme suit l'étape d'amélioration des sous-modèles non dominants (étape 2 de l'algorithme 9) ce qui garantit que $\Delta_{k+1} = \Delta_k$.
3. $\rho_k < \nu$, $\rho_k^j \geq \nu$ et tous les sous-modèles sont Λ-unisolvants : les hypothèses sur Δ_k impliquent que les conditions du lemme 5 sont remplies, ce qui implique que $\rho_k > \nu$. Une contradiction est levée, ce cas ne peut donc jamais se produire dans ces conditions.
4. $\rho_k < \nu$, $\rho_k^j < \nu$ et $F_k^j \neq \emptyset$: dans ce cas, l'étape de gestion de la région de confiance (étape 3(c)) entraine $\Delta_{k+1} = \Delta_k$.
5. $\rho_k < \nu$, $\rho_k^j < \nu$ et $F_k^j = \emptyset$: on a $\Delta_k < k_\Lambda \left\| g_k \right\|$, le lemme 4 implique donc que $C_k^j \neq \emptyset$ et par conséquent $\Delta_{k+1} = \Delta_k$.

Aucun de ces cas ne cause une diminution de la région de confiance. Une condition pour que la région de confiance se réduise est donc :

$$\Delta_k \geq \min\left((1-\nu)\frac{k_c}{k_{ef}n}, \frac{1}{k_H}, k_\Lambda, \mu\right) c$$

ce qui prouve donc le résultat avec $k_\Delta = \gamma \min\left((1-\nu)\frac{k_c}{k_{ef}n}, \frac{1}{k_H}, k_\Lambda, \mu\right) c$ □

On considère à présent le cas où le nombre d'itérations fructueuses est fini :

Lemme 7. *Supposons que seul un nombre fini d'itérations fructueuses ($\rho_k \geq \nu$) ne survienne. Alors*

$$\lim_{k\to+\infty} \inf \|g_k\| = 0$$

Démonstration. Sous les hypothèses précédentes, toutes les itérations deviennent infructueuses pour k grand : $\exists k_0 \geq 0$ tel que $\forall k \geq k_0$, $\rho_k < \nu$. Cela implique qu'à partir du rang k_0, les itérés x_k de l'algorithme deviennent invariants, c'est à dire $\forall k \geq k_0$, $x_{k+1} = x_k = x_*$.
Afin de trouver une contradiction, on suppose qu'il existe $c > 0$ tel que pour tout $k \geq k_0$, $\|g_k\| > c$. Le lemme 6 donne

$$\Delta_{k+1} > k_\Delta > 0.$$

Comme $\forall k \geq k_0$, $\rho_k < \nu$, la suite $(\Delta_k)_{k\geq k_0}$ est décroissante et minorée et est donc convergente. On appelle $\Delta_\infty = \lim_{k\to\infty} \Delta_k \geq k_\Delta$.
$\|g_k\|$ étant minoré par $c > 0$, il n'est pas possible pour l'algorithme de boucler infiniment dans l'étape critique (étape 2). De plus, les étapes dans lesquels Δ_k diminue entrainent :

$$\Delta_{k+1} = \gamma\Delta_k \text{ avec } \gamma < 1.$$

Puisque (Δ_k) converge vers une valeur strictement positive, l'algorithme ne peut pas boucler infiniment dans ces étapes. Les seuls étapes restantes dans l'algorithme sont :

- $\rho_k^j > \nu$ et il existe des sous-modèles non dominants et non Λ-unisolvants tels que $\rho_k^i < \nu$: comme x_k est constant pour $k \geq k_0$ et que le nombre d'étapes nécessaires à l'obtention d'un modèle Λ-unisolvant est fini, l'algorithme ne peut entrer qu'un nombre fini de fois dans cette étape.
- $\rho_k^j \leq \nu$ et $F_k^j \neq \emptyset$: l'algorithme ne peut entrer dans cette étape qu'un nombre fini de fois avant que tous les points d'interpolation de tous les sous-modèles ne soient dans $B(x_*, \Delta_\infty)$.
- $\rho_k^j \leq \nu$ et $C_k^j \neq \emptyset$. À chaque itération, l'algorithme remplace un point de Y_k^j tel que le polynôme de Lagrange associé possède la propriété $\left|l(x_k^+)\right| \geq \Lambda$, ce qu'il est impossible de faire indéfiniment.

On a levé une contradiction, on peut donc conclure que

$$\lim_{k\to+\infty} \inf \|g_k\| = 0.$$

□

Examinons à présent le cas où un nombre infini d'itérations fructueuses se présente :

Lemme 8. *Si un nombre infini d'itérations fructueuses arrive, on a*

$$\lim_{k\to+\infty} \inf \|g_k\| = 0.$$

Démonstration. De façon à lever une contradiction, on suppose qu'il existe $c > 0$ tel que pour tout $k \geq k_0$, $\|g_k\| > c$. Les hypothèses du lemme 6 sont remplies. Il existe donc $k_\Delta > 0$ tel que pour tout k suffisamment grand

$$\Delta_k > k_\Delta.$$

Pour toutes les itérations fructueuses, on a

$$F(x_k) - F(x_k^+) > \nu(m_k(x_k) - m_k(x_k^+)).$$

De plus, la condition de Cauchy (2.31) garantit que le point d'essai x_k^+ est choisi tel que

$$m_k(x_k) - m_k(x_k^+) > k_c \|g_k\| \min\left(\frac{\|g_k\|}{1 + \|H_k\|}, \Delta_k\right).$$

Par conséquent

$$F(x_k) - F(x_k^+) > \nu k_c c \min\left(\frac{c}{k_H}, k_\Delta\right) = k_d > 0.$$

En additionnant toutes ces inégalités, comme il y a un nombre infini d'itérations fructueuses, on peut conclure

$$\lim_{k\to\infty} F(x_k) = -\infty,$$

ce qui contredit le fait que F est minorée.
On a levé une contradiction, on peut donc en conclure que :

$$\lim_{k\to+\infty} \inf \|g_k\| = 0.$$

□

Pour pouvoir finaliser, on a besoin de deux résultats supplémentaires issus de [98] :

Lemme 9. *Pour chaque itération, on a*

$$\left|F(x_k^+) - m_k(x_k^+)\right| \leq \|\nabla F(x_k) - g_k\| \, \Delta_k + k_H \Delta_k^2.$$

Démonstration. Comme montré dans [98], pour que cette inégalité se réalise, on a seulement besoin que $F(x_k) = m_k(x_k)$ pour chaque itéré x_k. Cette propriété est assurée par l'algorithme 8. □

Lemme 10. *Supposons que*

(i) $g_k \neq 0$,

(ii) $\|\nabla F(x_k) - g_k\| \leq \frac{1}{2} k_c (1 - \nu) \|g_k\|$,

(iii) $\Delta_k \leq \frac{k_c}{2k_H}(1 - \nu) \|g_k\|$.

Alors l'itération k est fructueuse.

Démonstration. Avec le lemme précédent, ce résultat est une conséquence directe de la condition de Cauchy. □

Nous avons à présent tous les outils pour montrer le résultat final, en l'occurrence la convergence de la limite inf de $\|\nabla F(x_k)\|$ vers 0 (théorème 1) :

Démonstration. Afin de trouver une contradiction, supposons qu'il existe c_g et $k_0 \geq 0$ tels que

$$\forall k \geq k_0,\ \|\nabla F(x_k)\| \geq c_g.$$

Les deux lemmes 7 et 8 garantissent que quel que soit $\epsilon_q > 0$, il existe une itération k_q tel que $\left\|g_{k_q}\right\| < \epsilon_q$ au début de l'étape du test critique (étape 2). Cela implique qu'un nouveau jeu de sous-modèles est généré à l'itération k_q. Comme c'est vrai pour tout ϵ_q, les suites (k_q) et (ϵ_q) sont infinies et $\lim_{q\to\infty} \epsilon_q = 0$.
Considérons q suffisamment grand pour que

$$\epsilon_q \leq \frac{1}{2}\min\left(\frac{k_c(1-\nu)}{2k_g\Lambda}, \gamma\theta, \frac{\gamma k_c(1-\nu)}{2k_H}\right)c_g. \tag{2.33}$$

Après l'étape du test critique de l'itération k_q, le modèle m_{k_q} est Λ-unisolvant sur $B(x_{k_q}, \mu\epsilon_q)$. La proposition 5 entraine donc

$$\left\|\nabla F(x_{k_q}) - g_{k_q}\right\| \leq k_g\Lambda\mu\epsilon_q < k_g\Lambda\epsilon_q.$$

L'inégalité (2.33) ci-dessus implique alors

$$\left\|\nabla F(x_{k_q}) - g_{k_q}\right\| < k_g\Lambda\epsilon_q < \frac{1}{4}k_c(1-\nu)c_g < \frac{1}{4}k_c(1-\nu)\left\|\nabla F(x_{k_q})\right\|$$

et donc, a fortiori

$$\left\|\nabla F(x_{k_q}) - g_{k_q}\right\| \leq \frac{1}{2}\left\|\nabla F(x_{k_q})\right\|.$$

On a à présent l'inégalité

$$\left\|g_{k_q}\right\| = \left\|\nabla F(x_{k_q}) + g_{k_q} - \nabla F(x_{k_q})\right\| \geq \left\|\nabla F(x_{k_q})\right\| - \left\|\nabla F(x_{k_q}) - g_{k_q}\right\| \geq \frac{1}{2}\left\|\nabla F(x_k)\right\|,$$

ce qui implique

$$\left\|\nabla F(x_{k_q}) - g_{k_q}\right\| \leq \frac{1}{2}k_c(1-\nu)\left\|g_{k_q}\right\|.$$

Par conséquent, pour une telle valeur de q, il n'y a pas de boucle infinie dans l'étape du test critique. On a aussi, à la fin de l'étape du test critique de l'itération k_q,

$$\Delta_{k_q} = \theta\left\|g_{k_q}\right\| \geq \frac{1}{2}\theta c_g \geq \frac{\epsilon_q}{\gamma} > \epsilon_q.$$

Comme un nouveau modèle est généré par l'étape du test critique à l'itération k_q, on a $v_q = x_{k_q}$. De plus, la structure de l'algorithme entraine qu'après une telle génération, le modèle ne peut être modifié que si une itération fructueuse arrive ou que la région de confiance diminue trop ($\Delta_k < \epsilon_q$). L'algorithme peut diminuer la région de confiance pour les itérations $k \geq k_q$ jusqu'à

ce que $\Delta_k < \frac{k_c(1-\nu)c_g}{2k_H}$. Les hypothèses du lemme 10 sont alors vérifiées et une itération fructueuse $k_s \geq k_q$ arrive avec

$$\Delta_{k_s} \geq \frac{\gamma k_c(1-\nu)c_g}{4k_H} = \Delta_{lim} \geq \epsilon_q$$

Puisque le modèle n'a pas été modifié entre les itérations k_q et k_s, on a

$$\|g_{k_s}\| = \|g_{k_q}\| \geq \frac{1}{2}c_g$$

Puisque k_s est une itération fructueuse, le point d'essai $x_{k_s}^+$ est accepté en tant qu'itéré suivant x_{k_s+1} et

$$\frac{F(x_{k_s}) - F(x_{k_s+1})}{m_k(x_{k_s}) - m_k(x_{k_s+1})} > \nu$$

ce qui implique, avec la condition de Cauchy :

$$F(x_{k_s}) - F(x_{k_s+1}) > \nu \|g_{k_s}\| k_c \min\left(\frac{\|g_{k_s}\|}{1+\|H_{k_s}\|}, \Delta_{lim}\right) > \frac{1}{2}\nu c_g k_c \min\left(\frac{\frac{1}{2}c_g}{k_H}, \Delta_{lim}\right) > 0.$$

Ce scénario est répété pour chaque q tel que

$$\epsilon_q \leq \frac{1}{2}\min\left(\frac{k_c(1-\nu)}{2k_g\Lambda}, \gamma\theta, \frac{\gamma k_c(1-\nu)}{2k_H}\right)c_g,$$

ce qui entraine $\liminf_{k\to+\infty} F(x_k) = -\infty$. L'hypothèse que F est minorée est alors contredite. La conclusion suit :

$$\lim_{k\to+\infty} \inf \|\nabla F(x_k)\| = 0$$

□

2.5 Premiers résultats numériques avec l'algorithme DFO-PSOF

L'algorithme DFO-PSOF a été entièrement implémenté en C++ et intégré dans l'outil d'optimisation interne à IFP Énergies Nouvelles HUBopt (un diagramme de classe est présenté en annexe). Cette implémentation a permis de tester l'algorithme à la fois indépendamment sur des fonctions tests analytiques et sur des problèmes industriels de calage d'historique. On présente ici les premiers tests qui nous ont permis de valider numériquement cet algorithme.

2.5.1 Fonctions analytiques partiellement séparables

On montre ici les résultats numériques obtenus sur des fonctions analytiques partiellement séparables, en l'occurrence :

$$
\begin{aligned}
DQDRTIC(x) &= \sum_{i=1}^{p-2} (x_i^2 + 100x_{i+1}^2 + 100x_{i+2}^2) \\
LIARWHD(x) &= \sum_{i=1}^{p} 4(x_i^2 - x_1)^2 + (x_i - 1)^2 \\
BDQRTIC(x) &= \sum_{i=1}^{p-4} ((-4x_i + 3)^2 + (x_i^2 + 2x_{i+1}^2 + 3x_{i+2}^2 + 4x_{i+3}^2 + 5x_n^2)^2) \\
ARWHEAD(x) &= \sum_{i=1}^{p-1} ((x_i^2 + x_n^2)^2 - 4x_i + 3) \\
ROSENBROCK(x) &= \sum_{i=1}^{p-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2).
\end{aligned}
$$

Lorsque le nombre de variables p est suffisamment grand, chacune de ces fonctions est difficile à minimiser avec un algorithme d'optimisation sans dérivées classique. Cependant, elles s'écrivent toutes comme la somme de sous-fonctions qui dépendent d'un relativement petit nombre de paramètres. Le nouvel algorithme présenté ici est donc supposé donner de bons résultats comparativement aux méthodes classiques.

On teste d'abord la capacité de l'algorithme DFO-PSOF à traiter des problèmes en grande dimension : l'une des qualités souhaitée de cet algorithme est que sa vitesse de convergence dépend plus de la propriété de séparation de la fonction objectif que du nombre total de paramètres. La figure 2.6 montre le nombre d'évaluations de la fonction objectif nécessaire pour arriver à la convergence de l'algorithme en fonction de la dimension p du problème. On a choisi ici de stopper l'algorithme lorsque la région de confiance devient trop petite ($\Delta_k \leq 10^{-3}$). Il est clair sur cette figure que le nombre total de paramètres n'influe que marginalement sur le nombre d'évaluations nécessaire pour atteindre un optimum. Ce résultat valide numériquement les intuitions qui ont mené à la création de l'algorithme. Il est très important lorsqu'on considère l'application industrielle de calage d'historique : si on est capable de correctement identifier les séparations de la fonction objectif, on pourra en effet augmenter le nombre de paramètres sans accroitre le coût de calcul.

Les résultats donnés par DFO-PSOF ont également été comparés à ceux obtenus avec l'algorithme SQA [63]. SQA est un algorithme d'optimisation sans dérivées, ne prenant pas en compte la séparabilité de la fonction objectif, développé par les équipes IFPEN et est dans le cas sans contraintes très proche de l'algorithme NEWUOA de Powell [86] (voir algorithme 6). Ces résultats sont montrés dans les deux tableaux de la table 2.1. Le premier compare les performances des deux algorithmes lorsque les fonctions tests dépendent de 10 paramètres tandis que le second les compare pour des fonctions tests qui dépendent de 50 paramètres.

Ces tableaux montrent l'efficacité de l'algorithme DFO-PSOF sur ces fonctions tests : le nombre de points requis pour parvenir à la convergence de l'algorithme est bien inférieur lorsque la séparabilité partielle des fonctions est prise en compte. Ceci est particulièrement évident sur les fonctions dépendant de 50 paramètres, le nombre d'évaluations requis par DFO-PSOF étant systématiquement au moins un ordre de grandeur en dessous du nombre d'évaluations requis par SQA.

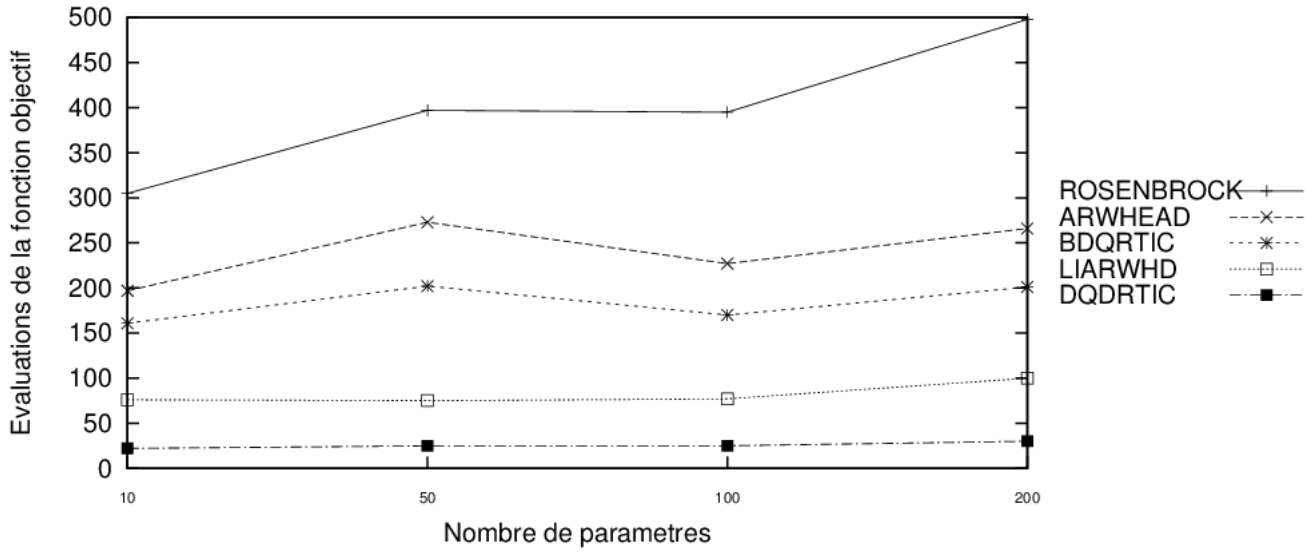


FIGURE 2.6 – Nombre d'évaluations de la fonction objectif pour arriver à convergence en fonction de la dimension du problème pour chacune des fonctions tests avec l'algorithme DFO-PSOF.

Le cas de la fonction DQDRTIC est très intéressant à analyser plus précisément. En effet, l'algorithme DFO-PSOF trouve l'optimum presque immédiatement quelle que soit la dimension du problème (entre 18 et 20 évaluations de la fonction objectif) tandis que SQA requiert beaucoup plus de calculs pour trouver la solution. En fait, la fonction étant elle-même quadratique, par définition de l'interpolation quadratique, si suffisamment de points sont présents dans l'ensemble d'interpolation, le modèle devient exact. Le fait que les tailles des ensembles d'interpolation ne soient pas fixes dans DFO-PSOF combiné au fait que chaque sous-fonction ne dépend que de 3 paramètres rend les modèles exacts en très peu de simulations : les sous-modèles sont exacts à partir du moment où chaque ensemble d'interpolation comprend $\frac{4*5}{2} = 10$ points. Quel que soit le nombre total de paramètres présents, l'algorithme convergera en un nombre presque constant d'évaluations. D'un autre côté, même avec un ensemble d'interpolation de taille variable, un algorithme d'optimisation classique n'obtiendra de modèle exact qu'après un beaucoup plus grand nombre de simulations (pour 50 paramètres, on a besoin de $\frac{51*52}{2} = 1326$ évaluations). La méthode SQA, comme NEWUOA, travaille avec un nombre fixe de points d'interpolations ($2p + 1$ points) et ne construit donc jamais de modèles exactes de la fonction objectif dans ce cas. Ceci explique que le nombre d'évaluations de la fonction objectif nécessaire à la convergence puisse être très supérieur à 1326.

2.5.2 Un cas simple de calage d'historique

L'algorithme DFO-PSOF est testé ici sur le cas décrit en section 1.4.1. On rappelle ci-dessous ses caractéristiques, la figure 2.7 montrant la carte de perméabilité du modèle de référence :

(i) 2500 m dans les directions x et y,

(ii) 10 m dans la direction z,

(iii) discrétisé uniformément par 50x50x1 mailles de 50 m dans les directions x et y et de 10 m dans la direction z,

(iv) 25 puits verticaux :
— 13 producteurs (PRO-1,PRO-3,...,PRO-25)

10 paramètres	SQA		DFO-PSOF	
Fonction	Valeur au meilleur point	Nombre d'itérations	Valeur au meilleur point	Nombre d'itérations
DQDRTIC	4,30E-12	839	7,30E-16	19
LIARWHD	3.51e-09	198	1,26E-09	48
BDQRTIC	18,2881	484	18,2880	198
ARWHEAD	3,19E-09	172	1,19E-09	37
ROSENBROCK	4,86E-09	618	9,20E-09	312
50 paramètres	SQA		DFO-PSOF	
Fonction	Valeur au meilleur point	Nombre d'itérations	Valeur au meilleur point	Nombre d'itérations
DQDRTIC	3,05E-13	4465	7,62E-16	18
LIARWHD	6,53e-09	521	1,90E-09	72
BDQRTIC	178,489	2651	178,489	210
ARWHEAD	8,83E-08	3220	6,70E-07	48
ROSENBROCK	4,05E-08	4057	1,32E-08	383

TABLE 2.1 – Comparaison des méthodes SQA et DFO-PSOF pour l'optimisation des fonctions décrites en début de section pour 10 paramètres (haut) et pour 50 paramètres (bas).

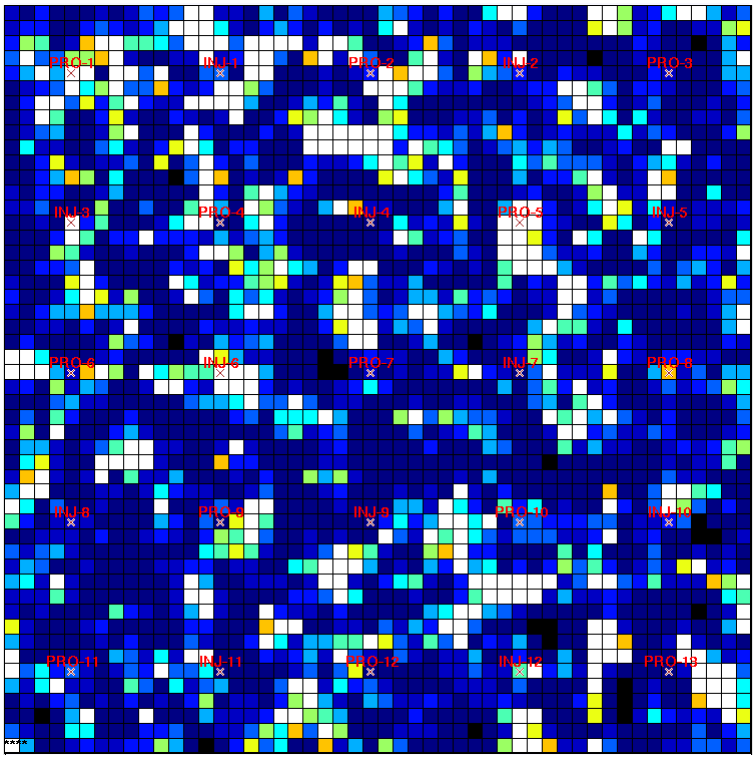


FIGURE 2.7 – Champ synthétique comprenant 13 puits producteurs et 12 puits injecteurs [37].

— 12 injecteurs (INJ-2,INJ-4,...,INJ-24)

(v) réservoir hétérogène : longueur de corrélation de 150 m dans les directions x et y,

(vi) pression initiale dans le réservoir de 250 bars,

(vii) pression en fond de puits imposée :

— 320 bars pour les injecteurs
— 180 bars pour les producteurs.

Un historique de données de production de 4000 jours est supposé connu.

On applique pour ce cas une paramétrisation par déformation graduelle généralisée décrite en section 1.2. 25 réalisations aléatoires indépendantes du modèle sont générées et la meilleure est choisie en tant que réalisation de base. On choisit de créer une zone autour de chaque puits et d'introduire pour chaque zone un paramètre de déformation graduelle locale. De façon à limiter le nombre total de paramètres, la taille des zones est fixée à 5 mailles de modèle (250m), ce qui fait donc un total de 25 paramètres à optimiser. La fonction objectif étant partiellement séparable, on considère qu'un paramètre n'a d'influence que sur les données de production du puits au centre de la zone qui lui est associée et sur les données de production des 4 puits les plus proches. Par exemple, comme illustré sur la figure 2.8, le paramètre associé à la zone autour du puits PRO_7 n'influence en vertu de cette supposition que les données de production des puits PRO_7, INJ_4, INJ_6, INJ_7 et INJ_9. Chaque sous-fonction objectif ne dépend donc au maximum que de 5 paramètres.

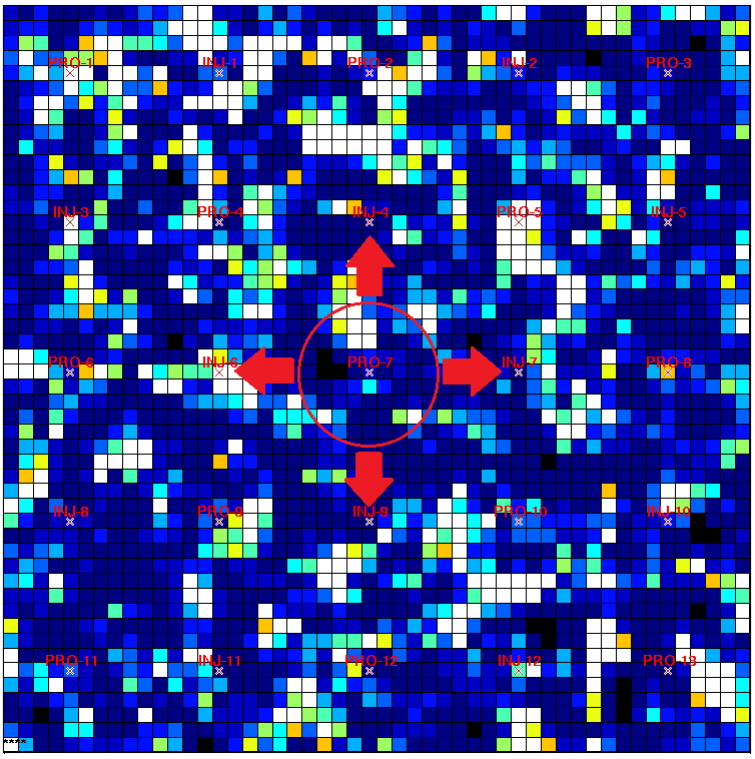


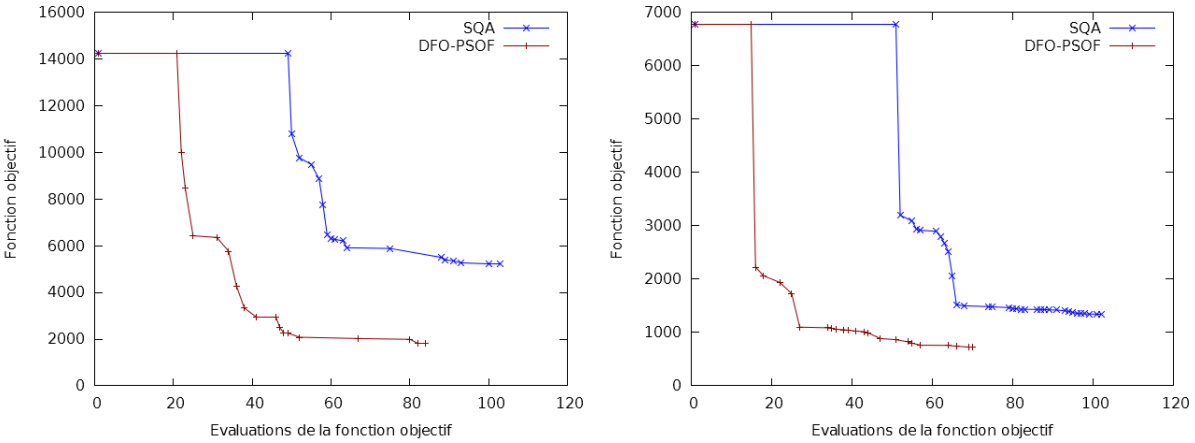
FIGURE 2.8 – Influence du paramètre associé au puits PRO_7.

On présente en figure 2.9 les résultats obtenus dans deux cas différents : dans chacun de ces cas, 25 réalisations indépendantes du modèle sont générées et les réalisations à combiner sont choisies en fonction des valeurs des fonctions objectifs locales. Les résultats des calages d'historique donnés par l'algorithme DFO-PSOF sont comparés avec ceux donnés par l'algorithme SQA [63]. Dans les deux cas, la nouvelle méthode DFO-PSOF donne plus rapidement de meilleurs résultats que SQA. Plus précisément :

(i) **dans le premier cas :**

— Pour obtenir la même valeur de fonction objectif que SQA en 105 itérations, DFO-PSOF ne requiert que 25 évaluations.
— La valeur de la fonction objectif à la fin de l'optimisation est de 5220.6 avec SQA et seulement de 1818.1 avec DFO-PSOF, soit un gain relatif de plus de 65%.

(ii) **dans le second cas :**
— Pour obtenir la même valeur de fonction objectif que SQA en 102 itérations, DFO-PSOF ne requiert que 27 évaluations.
— La valeur finale de la fonction objectif à la fin de l'optimisation est de 1340.2 avec SQA et seulement de 728.6 avec DFO-PSOF, soit un gain relatif de plus de 45%.



(a) Premier cas (b) Second cas

FIGURE 2.9 – Comparaison de calages d'historique par les méthodes SQA et DFO-PSOF.

Ces résultats valident l'hypothèse de séparabilité partielle dans ce cas synthétique simple. Par conséquent, comme on a pu le voir pour les cas analytiques, les modèles construits par le nouvel algorithme nécessitent moins de points d'interpolation que ceux construits avec la méthode SQA, ce qui implique une convergence plus rapide de DFO-PSOF en terme de nombre de simulations. Par ailleurs, la structure de la fonction objectif étant prise en compte, les modèles quadratiques locaux sont plus proches de la fonction objectif, ce qui peut constituer une explication au fait que les points finaux obtenus avec DFO-PSOF soient également meilleurs.

2.6 Bilan du chapitre

Dans cette section, on a cherché à exploiter lors de l'optimisation la structure particulière de la fonction objectif du problème de calage d'historique. En considérant le caractère local des paramètres de déformation graduelle généralisée, on peut supposer que la fonction objectif est partiellement séparable. Après une brève revue des méthodes d'optimisation sans dérivées utilisées dans le domaine, une méthode sans dérivées de type région de confiance utilisant cette structure nommée DFO-PSOF a été introduite. La convergence de cette méthode vers un point critique d'ordre un a été prouvée théoriquement et de premiers résultats numériques ont montré son efficacité tant sur des fonctions analytiques que sur un problème simple de calage d'historique.

Chapitre 3

Résultats numériques sur des modèles de réservoir

Ce chapitre est consacré aux tests des méthodes de paramétrisation et d'optimisation présentées dans les deux chapitres précédents sur des modèles de réservoir plus complexes : les cas PUNQ et Brugge. Ces deux réservoirs synthétiques sont couramment utilisés dans la littérature pour valider les nouvelles méthodes numériques dans le cadre d'une utilisation sur des problèmes réels. On commence par décrire les deux cas de réservoir en section 3.1. Plusieurs variantes de la méthode de paramétrisation présentée au chapitre 1 sont ensuite testées sur les deux cas tests en section 3.2. Enfin, après avoir identifié les indépendances entre paramètres et fonctions objectif locales, la méthode d'optimisation DFO-PSOF est appliquée en section 3.3.

3.1 Description des deux modèles de réservoir

3.1.1 Le cas PUNQ

Le cas PUNQ (Production forecasting with UNcertainty Quantification) est issu d'une étude de terrain réalisée par les partenaires industriels du projet PUNQ. Il est considéré comme un model de réservoir industriel de petite taille représentatif de la réalité du terrain. Le modèle est constitué de 19x25x5 mailles dont 1761 sont actives. Le réservoir modélisé est borné par une faille à l'est et au sud et par un aquifère au nord et à l'ouest. Six puits producteurs (PRO-1 à PRO-6) ainsi que sept puits injecteurs (INJ-1 à INJ-7) ont été forés. La carte de perméabilités de la couche supérieure du modèle de réservoir contenant les emplacements des puits est présentée en figure 3.1.

Les cartes de perméabilités et porosités ont été générées à l'aide du code FFTSIM [1] qui applique la méthode géostatistique "Fast Fourier Transform Moving Average" [89]. Ces cartes ont été générées indépendamment pour chacune des 5 couches du réservoir, les paramètres géostatistiques tels que la moyenne, la variance et les variogrammes sont choisis de façon à respecter au mieux le modèle géologique. Avec une réalisation de ce modèle géostatistique, un historique synthétique de production est généré sur 12 années à l'aide du simulateur PUMA-flow [2]. On définit la fonction objectif comme les sommes aux moindres carrés des historiques d'écart de production avec la réalisation de référence sur chaque puits producteur des données BHFP (Bottom Hole Flowing Pressure), WC (Water Cut) et GOR (Gas-Oil Ratio) et sur chaque puits injecteur

1. Code géostatistique développé à IFPEN, compris dans le logiciel Condor
2. Simulateur d'écoulements développé à IFPEN

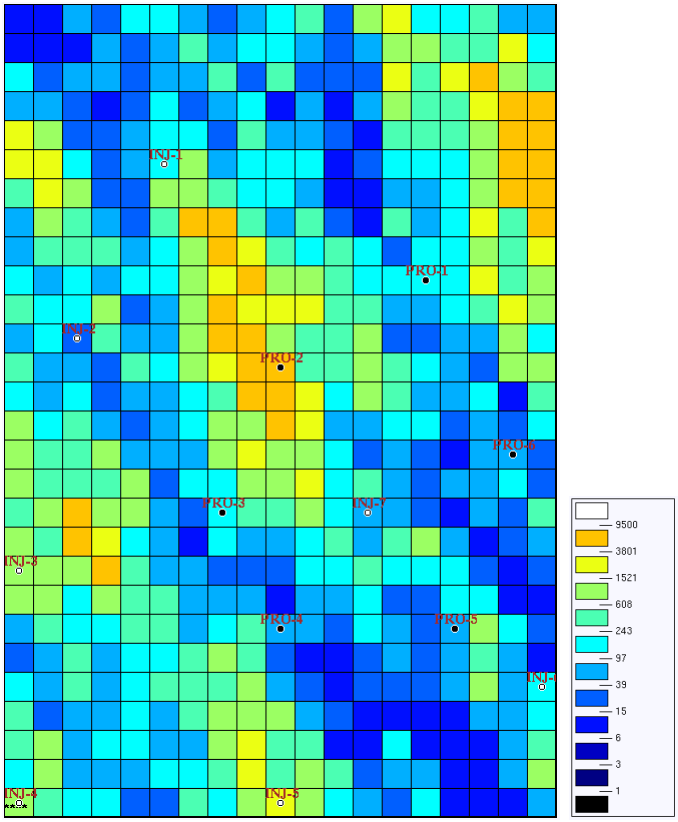


FIGURE 3.1 – Perméabilité de la première couche du réservoir PUNQ avec l'emplacement des puits.

des données BHFP. Dans une optique de calage d'historique, on va chercher à minimiser cette fonction objectif.

3.1.2 Le cas Brugge

Le cas Brugge est un modèle de réservoir synthétique construit en 2008 pour la conférence "Applied Technology Workshop" [81, 82]. Le modèle original possède 20 millions de mailles de taille moyenne 50m x 50m x 0.25m. On travaille dans ces tests sur une version mise à l'échelle ("upscaled") du modèle avec 60 000 mailles. Il comprend 30 puits sur le réservoir, 10 injecteurs et 20 producteurs. L'historique de production a été généré grâce au simulateur d'écoulement PUMA-flow sur des cartes de perméabilité et de porosité de référence. La figure 3.2 montre la carte de perméabilité de la première couche de la réalisation du modèle de référence. Nous disposons des données de production sur 10 ans. La fonction objectif d'écart avec la réalisation de référence est définie comme l'écart au sens des moindres carrés des historiques de production sur chaque puits producteur des données BHFP (Bottom Hole Flowing Pressure) et WC (Water Cut) et sur chaque puits injecteur des données BHFP.

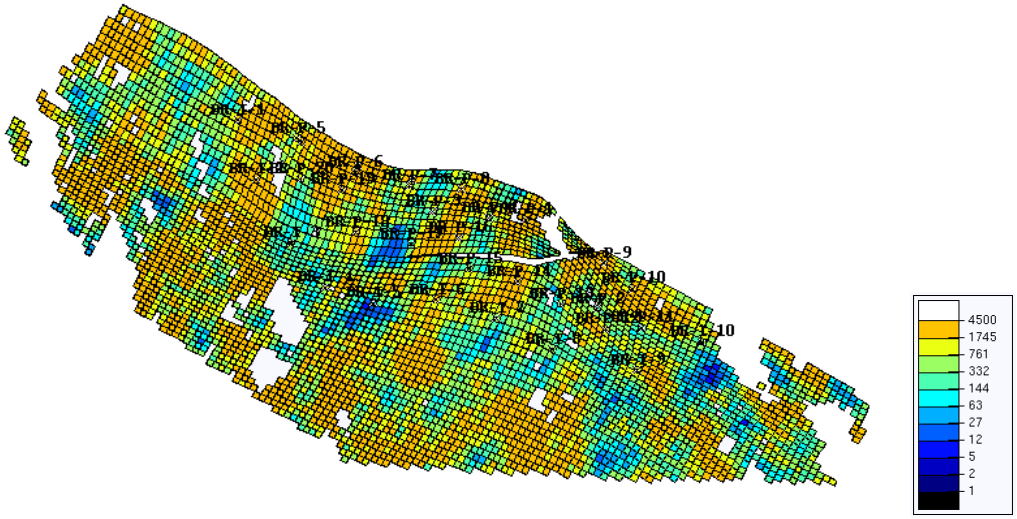


FIGURE 3.2 – Carte de perméabilité de la première couche du modèle de référence du cas Brugge.

3.2 Résultats numériques sur les méthodes de paramétrisation

3.2.1 Rappel de la méthodologie

On rappelle la méthodologie utilisée pour générer la réalisation de modèle initiale et définir les paramètres du problème de calage d'historique (voir chapitre 1). La méthode de patchwork améliorée est décrite comme suit :

1. Générer $N+1$ réalisations de modèle $(Z_0, ..., Z_N)$.
2. Appeler Z_0 le bruit blanc Gaussien associé à la réalisation de modèle présentant la plus faible valeur de fonction objectif.
3. Pour chaque puits i :
   — trier les valeurs des fonctions objectifs locales $f_i(Z_j)$. Soit Z_{j_i} le bruit blanc Gaussien tel que $f_i(Z_{j_i})$ soit minimale,
   — si $f_i(Z_0) > \alpha F(Z_0)$ et $f_i(Z_{j_i}) < f_i(Z_0)$, avec $0 < \alpha < 1$ une constante fixée au préalable, définir un zone paramétrée centrée autour du puits i et appliquer Z_{j_i} à l'intérieur. La zone peut par exemple être circulaire et paramétrée par son seul rayon r.
4. Effectuer une optimisation simple avec un unique paramètre contrôlant la taille de toutes les zones. Dans le cas de zones circulaires, on peut optimiser la taille des zones avec un unique paramètre de rayon.

Avec les zones ainsi créées, on peut tester plusieurs variantes de la méthode de déformation graduelle locale généralisée.

(i) Dans la première, on introduit pour chaque zone un paramètre de déformation graduelle locale et un paramètre contrôlant la forme de la zone.

(ii) Dans la seconde, on introduit toujours un paramètre de déformation graduelle locale pour chaque zone mais on n'introduit qu'un unique paramètre contrôlant simultanément la forme de toutes les zones.

(iii) Dans la troisième, les zones gardent leur forme initiale. Les seuls paramètres introduits sont alors ceux introduits par la méthode de déformation graduelle locale (un par zone).

Sur les deux cas testés, les zones créées sont des cercles de centre fixe et sont donc paramétrées par leur seul rayon. Dans le cas Brugge, les longueurs de corrélation de toutes les couches du réservoir sont isotropes (1000m dans les directions x et y), des zones circulaires sont donc bien adaptées au problème. Une certaine anisotropie est présente dans le cas PUNQ, mais chaque couche du réservoir présente un axe d'anisotropie différent : les azimuts sont respectivement de 30, 0, 45, -30 et 60 degrés par rapport à l'axe principal pour les cinq couches du réservoir. Il semble donc difficile de prendre en compte de façon efficace cette anisotropie à travers la forme des zones, c'est pourquoi des zones circulaires semblent aussi pouvoir convenir dans ce cas. Les longueurs de corrélations dans la direction principale d'anisotropie pour le cas PUNQ sont respectivement de 1001m, 750m, 1500m, 750m et 1250m dans les cinq couches du réservoir. Ces différences peuvent amener à penser que pour une zone donnée, considérer un unique rayon pour les cinq couches du réservoir pourrait poser problème. Cependant, les résultats numériques présentés ultérieurement montrent que ce n'est pas le cas.

3.2.2 Le cas PUNQ

3.2.2.1 Initialisation

Pour pouvoir appliquer la méthodologie de patchwork améliorée, nous avons généré 11 réalisations aléatoires et indépendantes du modèle de réservoir. Les valeurs des fonctions objectif locales peuvent être trouvées dans le tableau 3.1.

Réalisation	F	f_1 f_8	f_2 f_9	f_3 f_{10}	f_4 f_{11}	f_5 f_{12}	f_6 f_{13}	f_7
Z_0	649.92	77.36 0.44	0.61 32.32	168.09 0.50	0.47 137.22	88.77 0.40	0.61 0.48	142.54
Z_1	1512.56	135.45 0.64	0.73 19.51	232.71 0.26	0.38 1006.88	61.21 0.44	0.42 1.12	54.19
Z_2	1080.81	297.44 2.47	2.13 121.26	274.50 2.32	2.10 56.09	139.97 2.08	2.34 0.23	177.83
Z_3	882.13	128.77 1.36	1.42 86.12	312.55 1.27	1.32 28.82	124.96 1.08	1.41 0.43	192.57
Z_4	839.83	174.18 0.48	0.63 39.48	219.64 0.48	0.59 112.10	190.55 0.46	0.54 0.84	99.80
Z_5	974.12	224.49 0.98	1.15 79.80	262.93 0.92	1.05 120.98	148.73 0.93	1.00 0.65	130.45
Z_6	979.61	157.84 2.34	2.62 142.28	277.71 2.41	2.58 101.61	135.89 2.45	2.68 0.42	148.72
Z_7	1036.74	159.11 3.20	3.72 143.57	321.46 3.30	3.56 77.61	144.64 3.25	3.47 0.66	169.13
Z_8	682.45	73.26 0.44	0.57 75.58	261.14 0.41	0.49 9.99	138.65 0.38	0.50 0.56	120.43
Z_9	709.70	116.03 0.69	0.77 58.45	248.99 0.61	0.69 29.19	156.80 0.51	0.71 0.56	95.64
Z_{10}	934.26	95.68 0.58	0.73 48.12	196.17 0.15	0.28 403.52	101.26 0.49	0.51 0.86	85.79

TABLE 3.1 – Valeurs de la fonction objectif et des fonctions objectif locales pour chaque réalisation de modèle du cas PUNQ.

Dans ce tableau, la première fonction objectif locale f_1 est associée au puits PRO-1, la seconde à INJ-1, la troisième à PRO-2, etc.

L'application de la méthodologie de patchwork améliorée nous fait choisir Z_0 en réalisation de base (fonction objectif à 649.92). De même, 9 zones circulaires sont créées autour des puits PRO-1, PRO-3, PRO-4, PRO-5, PRO-6, INJ-2, INJ-3, INJ-5 et INJ-7 (voir figure 3.3).

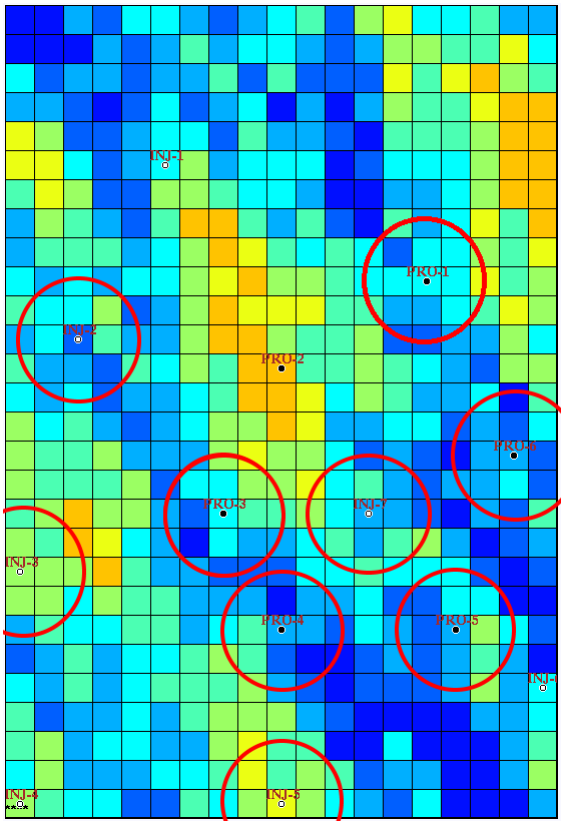


FIGURE 3.3 – Emplacement des zones créées par la méthode de patchwork améliorée.

On montre en figure 3.4 les variations de la fonction objectif en fonction de la taille des zones. On peut voir sur cette figure que choisir arbitrairement un rayon peut dégrader fortement la fonction objectif par rapport à la réalisation de base. Le choix d'un rayon de $720m$ pour toutes les zones ferait par exemple passer la fonction objectif de 649.92 à 828.20 alors que des rayons de $450m$ diminueraient la fonction objectif jusqu'à 572.58. Cependant, comme la fonction objectif varie de façon régulière avec le rayon des zones, on peut facilement trouver la valeur optimale du rayon avec un algorithme d'optimisation quelconque. La méthode d'optimisation SQPAL [33, 34] avec l'option Gauss-Newton donne pour optimum $R = 365.4m$ en sept évaluations de la fonction objectifs. Pour cette valeur de R la fonction objectif est égale à 502.032. Dans ce cas, il est donc possible de réduire la fonction objectif de plus de 22% pendant le processus de paramétrisation du problème.

Pour comparer la méthode de patchwork améliorée avec la méthode de patchwork classique utilisée en ingénierie de réservoir, un découpage du réservoir en zones délimitées par les mailles a été effectué (le découpage est présenté en figure 3.5). La réalisation de modèle correspondant au patchwork classique a alors été générée en considérant la même répartition des réalisations que précédemment : on associe la réalisation associée à Z_0 à l'extérieur des zones et la meilleure

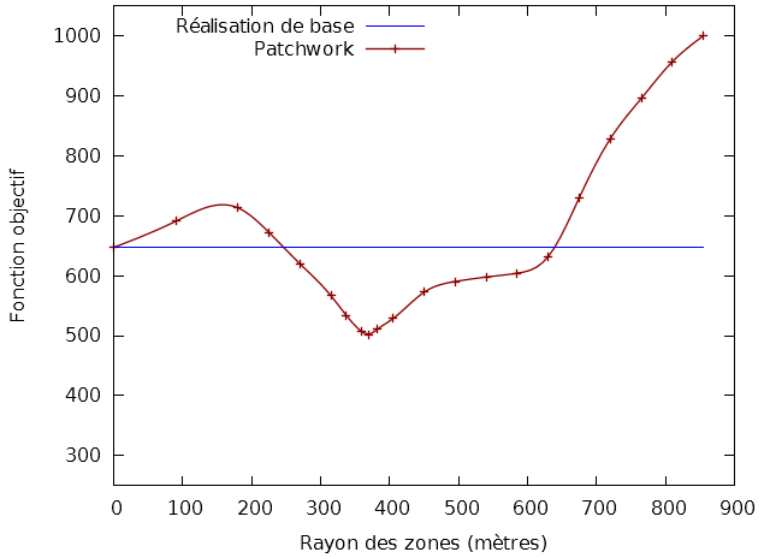


FIGURE 3.4 – Valeur de la fonction objectif après patchwork en fonction de la taille des zones pour le cas PUNQ.

réalisation locale à l'intérieur des zones. Après simulation de l'historique de production sur ce modèle, il s'avère que cette méthode détériore la fonction objectif. La fonction objectif passe en effet de 649.92 pour la réalisation associée à Z_0 à 851.12 pour la réalisation correspondant au patchwork classique. Ce résultat souligne l'importance de l'optimisation préliminaire de la forme des zones (étape 5 de la méthodologie de patchwork améliorée). Lorsque les zones ne sont pas parfaitement choisies, la fonction objectif dépend fortement de la taille des zones. Il est donc fréquent de choisir un mauvais point initial si on ignore toute considération sur la forme des zones.

3.2.2.2 Déformation graduelle généralisée

On commence par comparer la méthode de déformation graduelle généralisée avec la méthode de déformation graduelle locale classique. Pour cela, la réalisation de modèle obtenue par patchwork classique sur le découpage présenté en figure 3.5 étant moins bonne que la réalisation de base, on choisit comme réalisation initiale la réalisation de base et on introduit un paramètre de déformation graduelle pour chacune des zones créées précédemment. On présente en figure 3.6 le résultat des optimisations réalisées avec l'algorithme sans dérivées SQA [63] (algorithme dans dérivées proche de la méthode NEWUOA de Powell décrite dans l'algorithme 6 au chapitre 2). En bleu se trouve l'optimisation réalisée par déformation graduelle classique et en rouge celle réalisée par déformation graduelle généralisée. Dans ce cas, on a utilisé la variante de la déformation graduelle généralisée avec un unique paramètre de rayon.

L'optimisation débute pour les deux méthodes avec le modèle correspondant à la réalisation de base (meilleure parmi les dix générées initialement). La première marche qu'on peut observer sur la courbe de déformation graduelle généralisée correspond à l'optimisation initiale de la taille des zones et illustre le fait qu'on a besoin de sept évaluations de la fonction objectif pour trouver un meilleur point de départ. La méthode de déformation graduelle généralisée donne un bien meilleur résultat que la méthode classique : la valeur de la fonction objectif du modèle obtenu par déformation graduelle généralisée présente un gain relatif de plus de 48% sur la méthode de déformation graduelle classique.

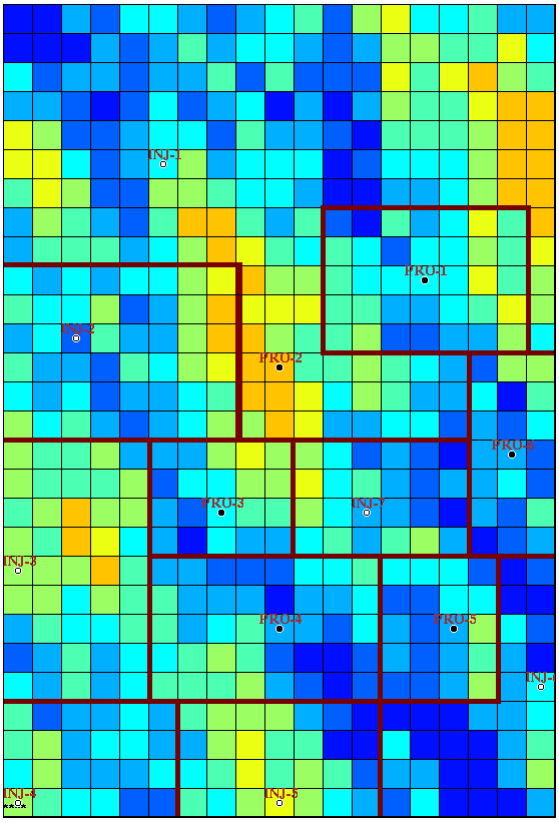


FIGURE 3.5 – Découpage du modèle de réservoir pour un patchwork classique.

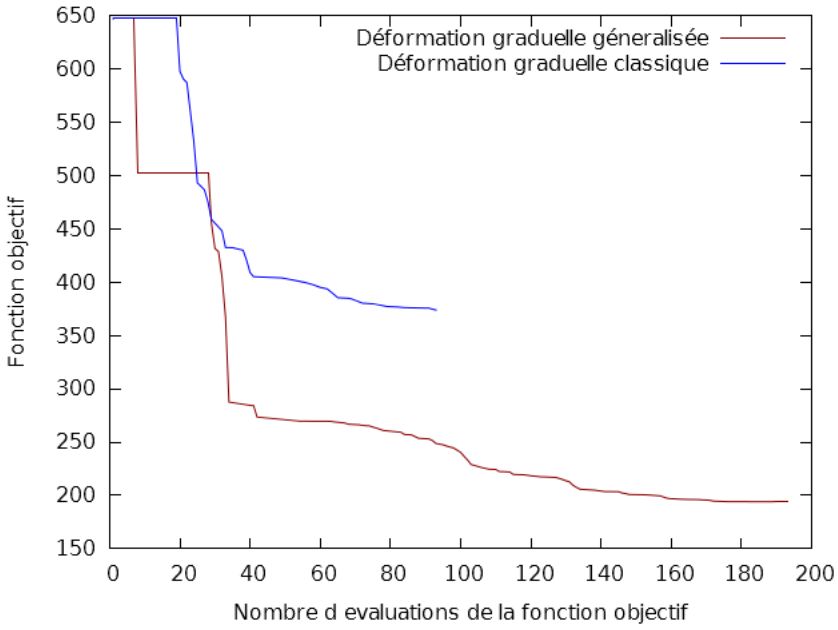


FIGURE 3.6 – Optimisation de la fonction objectif par déformation graduelle classique et par déformation graduelle généralisée sur le cas PUNQ.

Afin de comprendre plus précisément l'impact de la prise en compte de la forme des zones sur la décroissance de la fonction objectif, nous avons comparé les performances des trois variantes de la méthode de déformation graduelle généralisée. Les résultats de trois calages d'historique réalisés sans prendre en compte de rayon (courbe rouge), avec un unique paramètre de rayon pour toutes les zones (courbes bleue) et un paramètre de rayon indépendant pour chaque zone (courbe verte) sont présentés en figure 3.7. L'algorithme SQA [63] a également été utilisé dans ce cas pour les trois optimisations.

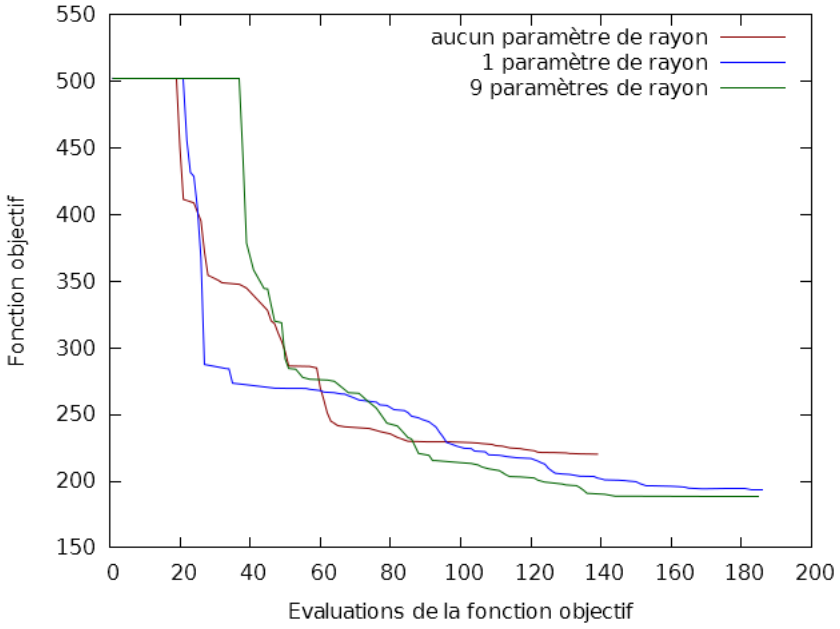


FIGURE 3.7 – Optimisation de la fonction objectif pour les 3 variantes de la méthode de déformation graduelle généralisée.

Plusieurs observations sont à noter sur ces résultats : premièrement, il est intéressant de voir que l'introduction de paramètres supplémentaires n'influe pas ou presque sur la vitesse de convergence. La méthode avec un paramètre de rayon fait diminuer marginalement plus vite la fonction objectif en début d'optimisation mais est rattrapée rapidement par les deux autres. De plus, les méthodes autorisant à faire varier les valeurs des rayons permettent au final de diminuer la fonction objectif de façon significativement plus importante (écart relatif de 15% à la fin des optimisations) en un nombre similaire de simulations. On peut déduire de ces deux observations que la possibilité de faire varier les valeurs des rayons des zones durant l'optimisation apporte une amélioration significative sans altérer le nombre de simulations nécessaires. D'un autre côté, il n'y a qu'un très faible écart entre les deux méthodes impliquant des paramètres permettant de modifier la taille des zones : malgré l'introduction de huit paramètres supplémentaires, le gain apporté par la méthode considérant tous les rayons indépendamment n'est que marginal (de l'ordre de 2%). On peut apporter une explication à ce résultat en considérant les valeurs des rayons optimaux données par les deux méthodes pour chaque zone. En effet, à l'optimum, l'un des rayons est nul (rayon de la cinquième zone). L'optimiseur est donc probablement tombé dans un minimum local, sans possibilité d'en ressortir. On peut en conclure que, s'il est avantageux de ne pas négliger les rayons, il n'est pas forcément nécessaire de les considérer individuellement sur le cas PUNQ, puisque l'ajout de nouveaux paramètres a tendance à créer de nouveaux optima locaux dans ce test.

3.2.3 Le cas Brugge

3.2.3.1 Méthode de Patchwork améliorée

De la même façon que dans les tests précédents, on débute la méthodologie de patchwork améliorée par la génération de 19 réalisations aléatoires et indépendantes du modèle de réservoir. On donne les valeurs des fonctions objectif et fonctions objectif locale pour chacune des réalisations dans le tableau 3.2. Les fonctions objectifs locales du tableau correspondent respectivement aux puits I-1,...,I-10,P-1,...,P-20.

	F	I-1 P-1 P-19	I-2 P-10 P-2	I-3 P-11 P-20	I-4 P-12 P-3	I-5 P-13 P-4	I-6 P-14 P-5	I-7 P-15 P-6	I-8 P-16 P-7	I-9 P-17 P-8	I-10 P-18 P-9
Z_0	68.94	0.003 0.297 31.22	0.015 3.993 7.221	0.014 0.797 0.716	0.006 0.867 0.244	0.004 1.194 0.207	0.008 0.742 14.21	0.008 1.616 0.402	0.013 3.041 0.284	0.010 0.090 0.548	0.007 0.345 0.803
Z_1	98.72	0.122 0.061 13.25	0.086 41.075 3.738	0.067 15.939 3.774	0.090 1.674 0.061	0.100 0.526 0.062	0.095 7.377 1.085	0.103 3.931 0.181	0.107 1.416 0.065	0.112 1.333 0.070	0.112 1.659 0.434
Z_2	106.78	0.093 0.032 1.305	0.049 30.73 0.244	0.047 25.49 1.949	0.048 0.706 0.103	0.075 14.06 0.092	0.054 6.367 10.79	0.077 4.536 4.241	0.078 0.758 0.108	0.094 2.589 0.043	0.073 0.371 1.560
Z_3	97.69	0.069 0.096 11.71	0.035 1.930 4.731	0.039 2.393 0.592	0.049 0.272 0.061	0.064 1.357 0.222	0.049 4.577 35.41	0.044 6.810 0.417	0.053 20.42 0.054	0.055 2.385 0.036	0.048 3.565 0.126
Z_4	69.77	0.015 0.275 0.814	0.017 1.521 0.657	0.002 2.067 8.160	0.012 3.503 0.155	0.022 0.587 0.271	0.016 3.774 21.78	0.019 5.308 4.725	0.024 1.828 0.167	0.026 13.14 0.262	0.031 0.482 0.104
Z_5	140.43	0.111 0.089 13.10	0.073 1.168 0.515	0.069 17.94 0.550	0.090 0.126 0.041	0.112 0.372 0.260	0.086 2.183 38.78	0.087 1.820 29.31	0.096 25.33 0.052	0.103 1.966 0.033	0.104 5.664 0.179
Z_6	115.73	0.067 0.036 9.384	0.049 4.594 0.408	0.042 2.290 0.195	0.050 7.471 0.036	0.077 0.331 0.049	0.063 3.784 74.19	0.073 0.586 1.245	0.089 6.053 0.047	0.080 3.903 0.027	0.077 0.303 0.112
Z_7	104.73	0.055 0.038 57.84	0.040 2.973 0.438	0.030 1.148 0.760	0.038 2.198 0.054	0.062 0.269 0.014	0.039 0.212 6.410	0.068 9.743 0.024	0.036 0.26 0.016	0.050 0.233 0.071	0.062 1.120 0.429
Z_8	203.10	0.189 0.671 0.336	0.195 7.423 25.01	0.243 6.741 8.618	0.222 2.558 0.593	0.187 2.705 0.643	0.204 4.016 34.26	0.218 10.55 0.297	0.197 55.84 0.493	0.191 5.95 0.664	0.187 0.486 33.16
Z_9	177.28	0.184 0.063 5.278	0.134 9.252 7.335	0.109 4.355 2.341	0.138 2.838 0.180	0.173 1.579 0.089	0.150 2.888 56.09	0.178 11.87 33.53	0.201 24.00 0.091	0.189 0.921 0.057	0.217 2.495 10.32
Z_{10}	253.21	0.041 0.556 6.006	0.059 22.83 17.15	0.040 20.58 2.717	0.052 3.704 2.862	0.040 2.966 0.522	0.055 3.422 45.78	0.048 8.007 56.81	0.046 55.18 0.295	0.052 1.778 0.390	0.038 0.341 0.813

Z_{11}	88.54	0.005	0.005	0.001	0.000	0.002	0.000	0.000	0.000	0.000	0.000
		0.139	26.89	1.891	2.311	6.246	1.455	0.665	11.11	2.717	1.531
		7.683	0.443	3.302	0.217	0.195	7.992	13.225	0.091	0.243	0.157
Z_{12}	141.16	0.005	0.004	0.012	0.013	0.006	0.013	0.012	0.015	0.007	0.011
		0.016	8.075	9.023	0.627	0.170	0.898	10.019	2.424	9.173	0.479
		7.685	1.767	1.278	0.055	0.023	73.50	15.18	0.073	0.050	0.528
Z_{13}	92.88	0.035	0.008	0.007	0.013	0.024	0.011	0.011	0.009	0.029	0.010
		0.134	0.946	1.991	1.434	4.747	1.529	1.928	46.77	15.03	2.254
		0.454	0.835	1.391	0.090	0.015	12.80	0.015	0.056	0.165	0.114
Z_{14}	109.07	0.001	0.000	0.000	0.000	0.002	0.000	0.001	0.001	0.001	0.001
		0.019	29.90	3.615	0.083	0.356	1.544	8.444	18.90	2.322	0.265
		0.240	12.47	2.351	0.980	0.023	14.83	8.298	3.253	0.064	1.082
Z_{15}	118.51	0.047	0.011	0.022	0.025	0.037	0.025	0.015	0.016	0.032	0.020
		0.003	9.437	3.464	0.526	1.332	0.117	7.404	11.17	3.732	1.152
		1.827	3.051	11.90	0.020	0.003	44.83	17.62	0.007	0.116	0.522
Z_{16}	82.52	0.035	0.016	0.014	0.023	0.034	0.018	0.029	0.024	0.035	0.028
		0.042	3.287	0.269	1.756	1.439	3.347	2.768	0.627	3.491	0.174
		4.012	0.212	0.267	0.430	0.019	49.18	10.74	0.136	0.044	0.016
Z_{17}	160.06	0.085	0.071	0.062	0.073	0.091	0.071	0.067	0.065	0.092	0.084
		0.024	2.792	8.253	0.129	8.127	2.935	7.526	10.86	7.253	2.786
		82.15	1.612	0.410	0.151	0.027	19.57	4.174	0.031	0.297	0.163
Z_{18}	235.58	0.001	0.011	0.009	0.005	0.002	0.004	0.006	0.002	0.002	0.002
		0.142	8.712	6.012	0.639	1.477	0.073	2.181	83.036	10.67	1.608
		23.48	29.056	7.881	0.078	0.105	58.75	1.460	0.040	0.105	0.004

TABLE 3.2 – Valeurs de la fonction objectif et des fonctions objectif locales pour chaque réalisation de modèle du cas Brugge.

La seconde étape de la méthodologie de patchwork améliorée donne Z_0 pour réalisation de base (fonction objectif de 68.94). Étant donnés le grand nombre de puits creusés sur le réservoir et les fortes disparités qu'on peut observer entre les différentes fonctions objectif locales, on va chercher dans ce test à limiter le nombre total de paramètres du problème en modifiant légèrement l'étape 4 de la méthodologie de patchwork. Plutôt que de créer une zone chaque fois que la réalisation de base n'est pas localement la meilleure, on restreint la création de zone au cas où la différence entre la fonction objectif locales de la meilleure réalisation locale et celle de la réalisation de base est suffisante. Plus précisément, pour chaque puits, on créé une zone si :

— il existe localement une réalisation de modèle associée à un bruit blanc Gaussien Z_k meilleure que celle associée à Z_0

— la différence entre les fonctions objectif locales associées à Z_0 et à Z_k est supérieure à ϵ.

Avec cette façon de choisir les zones, on s'assure qu'on n'agira que sur les régions du réservoir présentant un fort potentiel d'amélioration. Dans ce test, on choisit $\epsilon = 1$. Ainsi, sept zones circulaires sont créées autour des puits $P-10$, $P-13$, $P-15$, $P-16$, $P-19$, $P-2$ et $P-5$. Dans ces zones, on combinera respectivement les réalisations Z_{13}, Z_{12}, Z_6, Z_7, Z_{14}, Z_{16} et Z_1 avec la réalisation de base Z_0. On montre la répartition géographique des zones sur la figure 3.8.

La figure 3.9 montre la variation de la fonction objectif avec le rayon des zones lorsqu'un patchwork est appliqué sur ces zones. À nouveau, un choix arbitraire de rayon de zones aurait le potentiel de détériorer la valeur de la fonction objectif par rapport à la réalisation de base. Un choix de rayon de $100m$ ferait par exemple passer la fonction objectif de 68.94 pour la réalisation de base à 104.32. Heureusement, comme le montre la figure 3.9, les variations de

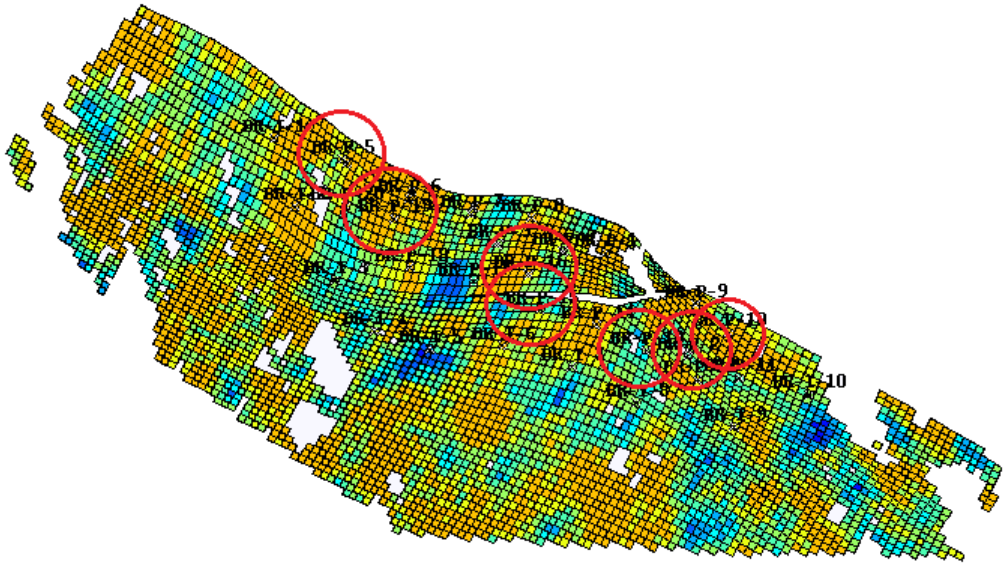


FIGURE 3.8 – Répartition des zones créées par la méthode de patchwork améliorée sur le réservoir Brugge.

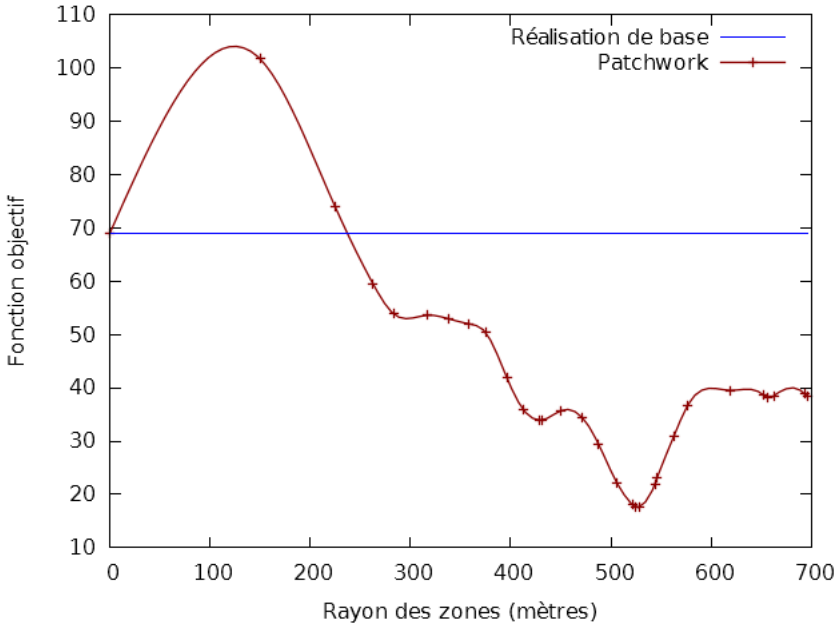


FIGURE 3.9 – Évolution de la valeur de la fonction objectif en fonction de la taille des zones sur le réservoir Brugge.

la fonction objectif avec le rayon des zones sont très régulières, il est donc facile de trouver le point minimal de cette courbe en peu de simulations. La méthode d'optimisation sans dérivées SQA donne un optimum égal à $525m$ en 10 simulations d'écoulement. La méthode de patchwork permet ainsi au cours de la procédure de paramétrisation du problème de faire passer la fonction objectif de 68.9 à 17.5, soit un gain relatif de plus de 74%.

Pour évaluer les performances de la méthode de patchwork améliorée, on propose en figure 3.10 un découpage classique en ingénierie de réservoir des zones suivant les mailles du domaine. La valeur de la fonction objectif de la réalisation de modèle obtenue avec ce découpage est de

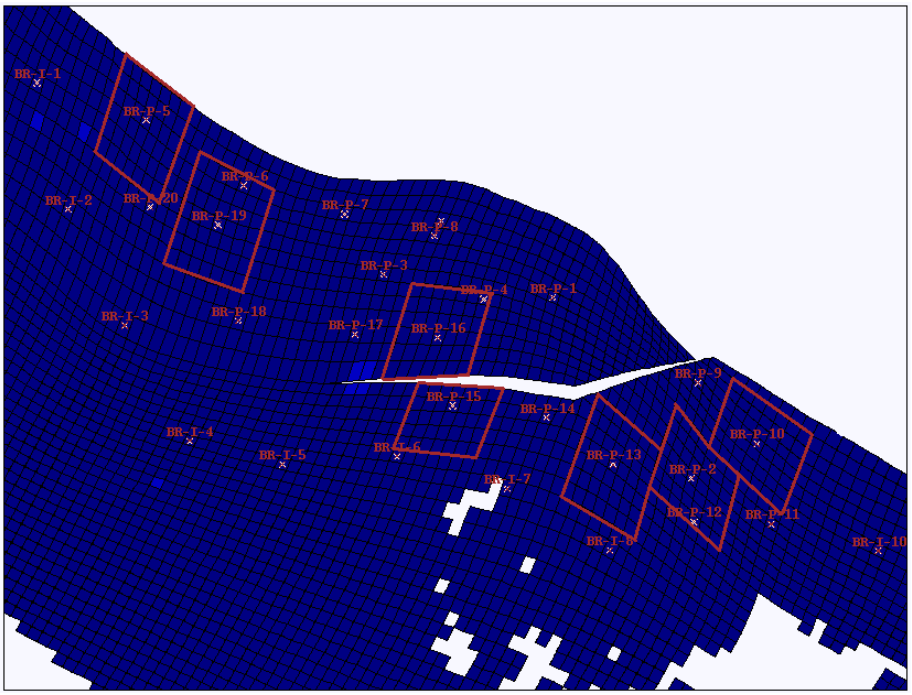


FIGURE 3.10 – Découpage classique du réservoir Brugge en zones délimitées par les mailles du modèle.

45.21. Dans ce cas, le patchwork permet toujours d'améliorer la fonction objectif (la fonction objectif est de 68.94 pour la réalisation de base), mais est loin du point obtenu avec des tailles de zones flexibles (patchwork amélioré). De plus, le fait que cette méthode ne permette pas de modifier les zones implique que la décroissance de la fonction objectif sera limitée si l'une des zones a été initialement mal définie.

3.2.3.2 Déformation graduelle généralisée

Comme pour le cas PUNQ, une comparaison entre la méthode de déformation graduelle classique et la méthode de déformation graduelle généralisée a été effectuée. Pour cela, on teste la méthode de déformation graduelle classique en reprenant le découpage classique présenté en figure 3.10 et en introduisant un paramètre de déformation graduelle par zone. Deux calage d'historique ont été réalisés par déformation graduelle classique en considérant comme réalisation de départ la réalisation obtenue par patchwork et la réalisation de base. La méthode de déformation graduelle généralisée est quant à elle testée en considérant les sept zones circulaires présentées en figure 3.8 et en introduisant un paramètre de déformation graduelle et un paramètre de rayon pour chacune des zones. On présente en figure 3.11 les résultats des trois optimisations réalisées. La première marche qu'on peut observer sur la courbe de déformation graduelle généralisée correspond à l'optimisation initiale de la taille des zones et illustre le fait qu'on a besoin de 10 évaluations de la fonction objectif pour trouver un meilleur point de départ. Dans ce cas, on peut observer que l'optimisation initiale sur la taille des zones est suffisante pour obtenir un meilleur résultat que la méthode de déformation graduelle classique, quelque que soit le point

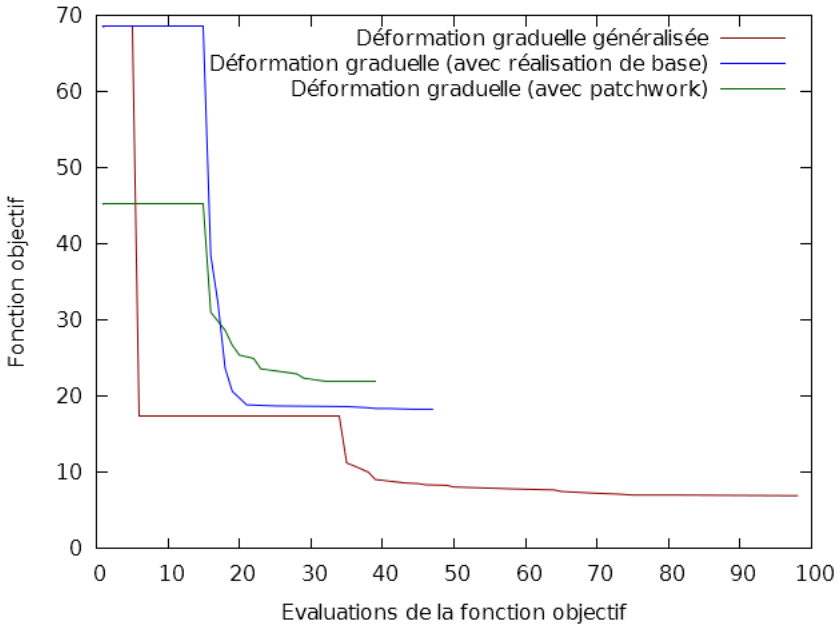


FIGURE 3.11 – Optimisation de la fonction objectif en utilisant la méthode de déformation graduelle classique avec la réalisation de base comme point initial (en bleu), avec la réalisation obtenue par patchwork (en vert) et la méthode de déformation graduelle généralisée (en rouge) sur le cas Brugge.

de départ choisi. La valeur optimale de la fonction objectif obtenue avec la méthode classique est de 18.30 tandis que celle donnée par la méthode généralisée est de 7.18, soit un gain relatif de plus de 60%. Il est donc évident que sur le cas Brugge, le choix des zones est critique pour une bonne décroissance de la fonction objectif.

Afin d'analyser le gain apporté par la méthode de déformation graduelle généralisée, on peut étudier la fonction objectif locale au puits producteur P-2. Après optimisation, la fonction objectif locale vaut 3.51 pour la déformation graduelle classique et 0.16 pour la déformation graduelle généralisée, ce qui indique un bien meilleur calage des données pour le puits P-2 avec la nouvelle méthode de paramétrisation. Plus précisément, la méthode de déformation graduelle locale a permis un bon calage des données de production BHFP et GOR mais ne donne pas un modèle approchant de façon satisfaisante le modèle de référence sur ce puits. On présente en figure 3.12 les historiques de Water Cut (WC) du puits producteur P-2 simulés par les modèles obtenus via déformation graduelle classique et déformation graduelle généralisée. En bleu se trouve l'historique de référence, en rouge celui simulé après déformation graduelle classique et en vert celui simulé après déformation graduelle généralisée. Cette figure montre que la méthode de déformation graduelle généralisée a pu éliminer une partie du problème que posait la déformation graduelle classique et offre un bien meilleur calage du WC sur le puits P-2.

Pour approfondir l'étude de l'impact des zones sur la décroissance de la fonction objectif, nous avons aussi comparé sur le cas Brugge les trois variantes de la méthode de déformation graduelle généralisée (0, 1 et 7 paramètres de rayon en addition des 7 paramètres de déformation graduelle, les rayons de départ étant toujours de $525m$). La figure 3.13 présente les résultats obtenus.

On peut immédiatement remarquer que dans ce cas, l'ajout d'un paramètre de rayon n'améliore pas la diminution de la fonction objectif. La valeur optimale du rayon trouvée par l'optimiseur est de $520.5m$, très proche de la valeur $525m$ choisie initialement. On peut en conclure que

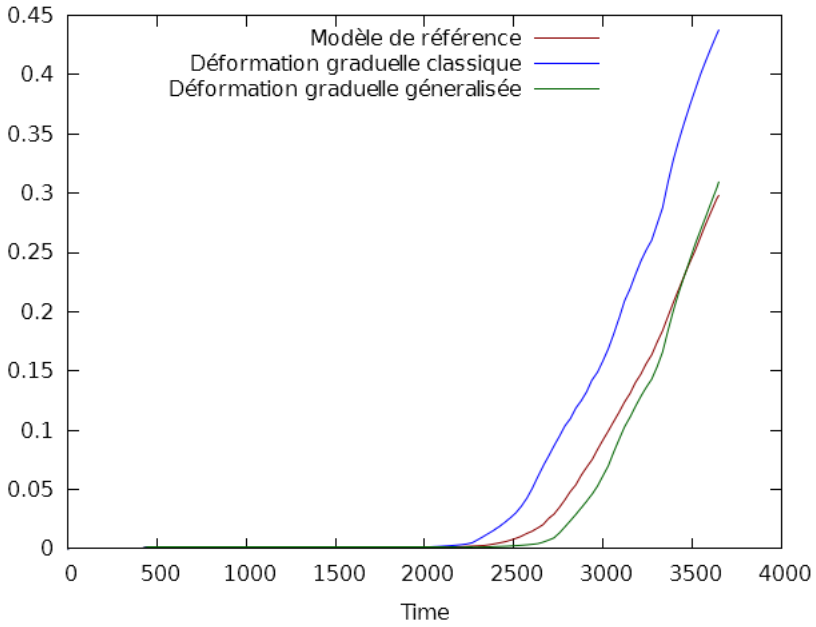


FIGURE 3.12 – Calage du Water Cut (WC) sur le puits P-2 après optimisation par déformation graduelle classique et généralisée, cas Brugge.

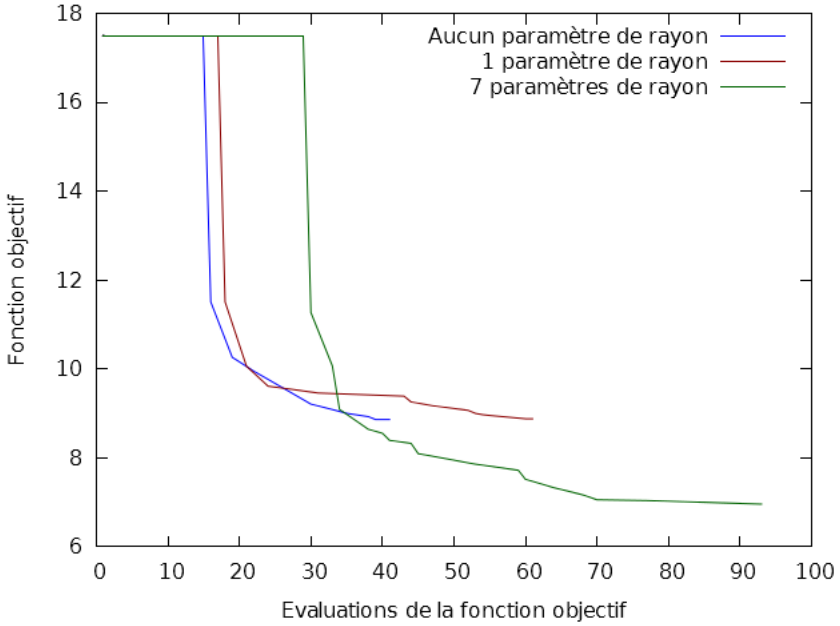


FIGURE 3.13 – Optimisation de la fonction objectif par les trois variantes de la méthode de déformation graduelle généralisée (valeur initiale des rayons à 525m), cas Brugge.

dans ce cas, l'optimisation initiale du rayon est suffisante et que l'ajout d'un paramètre de rayon unique n'est pas nécessaire pour une décroissance maximale de la fonction objectif. D'un autre côté, l'optimisation avec 7 rayons de zones variables et indépendants améliore significativement la fonction objectif : la valeur de la fonction objectif après optimisation avec 7 rayons est de 7.18 tandis que celle obtenue sans paramètre de rayon est de 8.86, soit un gain relatif de plus

de 21%. Les tableaux suivants (table 3.3) donnent respectivement les valeurs des paramètres de rayon et de déformation graduelle pour la meilleure simulation.

Rayon 1	Rayon 2	Rayon 3	Rayon 4	Rayon 5	Rayon 6	Rayon 7
612	413	562	532	533	533	523

DG 1	DG 2	DG 3	DG 4	DG 5	DG 6	DG 7
0.50	0.77	0.94	0.31	0.51	0.42	0.26

TABLE 3.3 – Valeurs des paramètre de rayon (en mètres) et de déformation graduelle à la fin de l'optimisation (valeur initiale des rayons à $525m$).

On constate que seuls les rayons des deux premières zones varient significativement ($612m$ et $413m$ respectivement), les autres restant relativement proches de leur valeur initiale de $525m$. Dans ce cas, l'ajout des rayons indépendants permet d'identifier deux zones non idéalement définies et de les améliorer pour une meilleure décroissance de la fonction objectif.

Optimiser la taille des zones est donc toujours important pour obtenir une bonne décroissance de la fonction objectif. Cependant, la plupart des rayons étant initialement biens choisis, il est nécessaire de considérer tous les rayons de façon indépendante pour pouvoir en tirer parti.

3.2.3.3 Importance du prétraitement

Les mêmes tests des trois variantes de la déformation graduelle généralisée que dans la section précédente ont été réalisés en choisissant pour réalisation initiale une réalisation de modèle construite par patchwork améliorée en fixant arbitrairement le rayon de toutes les zones à $225m$. On présente les résultats d'optimisation en figure 3.14.

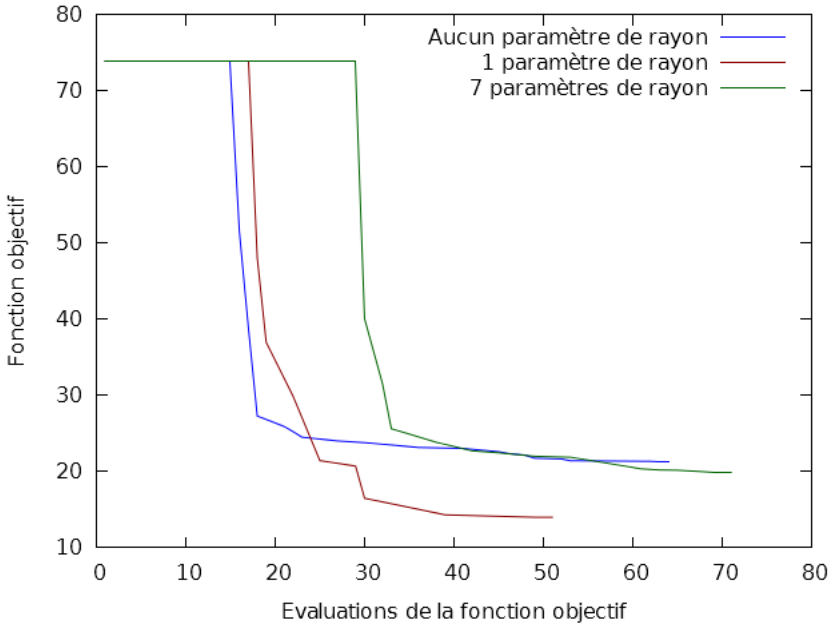


FIGURE 3.14 – Optimisation de la fonction objectif par les 3 variantes de la méthode de déformation graduelle généralisée (valeur initiale des rayons à $225m$), cas Brugge.

La valeur initiale de la fonction objectif est nettement moins bonne que dans le cas où la méthodologie de patchwork a été appliquée dans son ensemble. En effet, sans correction sur

la valeur initiale des rayons des zones, il n'est pas possible d'assurer que la modification de la réalisation du modèle via création de zones ne détériorera pas le modèle. Dans ce cas, le fait de choisir arbitrairement le rayon des zones à $225m$ mailles donne une valeur de la fonction objectif de 73.87. Cette valeur est largement moins bonne que celle obtenue après optimisation des rayons (17.50) ou même que celle donnée par la réalisation de base (68.52). Là où une simple optimisation à un paramètre permettait déjà d'atteindre une diminution de la fonction objectif de plus de 74%, choisir arbitrairement le rayon fait partir la méthode avec un handicap de 7% sur la fonction objectif.

De plus, un paramètre de rayon permet de diminuer grandement la valeur de la fonction objectif : 13.99 pour la valeur optimale obtenue avec un paramètre de rayon contre 21.28 sans considérer de rayon, soit un gain relatif de plus de 34%. Le rayon initial n'étant pas bien choisi, lui permettre de varier lors de l'optimisation apporte un gros bénéfice. Il est intéressant de noter que la valeur du rayon trouvée en fin d'optimisation est de $523m$, très proche de la valeur $525m$ trouvée grâce à l'optimisation initiale des rayons. L'introduction d'un rayon en paramètre du problème d'optimisation permet en fait de rattraper une partie du retard induit par l'absence de prétraitement.

Le dernier point à noter sur ces résultats est le fait que l'introduction de sept rayons indépendants en tant que paramètres du problème ne permet pas l'amélioration des résultats. Les rayons initiaux étant très éloignés des rayons optimaux, il est très difficile pour l'optimiseur de perturber suffisamment les paramètres pour parvenir à une solution satisfaisante sans tomber dans un minimum local. C'est ce qu'on observe ici, les valeurs des rayons optimaux sont donnés dans le tableau suivant (table 3.4) :

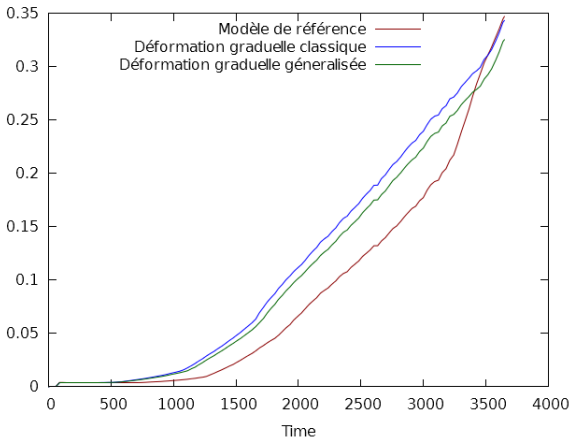
Rayon 1	Rayon 2	Rayon 3	Rayon 4	Rayon 5	Rayon 6	Rayon 7
232	247	242	148	391	227	320

TABLE 3.4 – Valeurs des rayons (en mètres) après optimisation de la fonction objectif (valeur initiale à $225m$), cas Brugge.

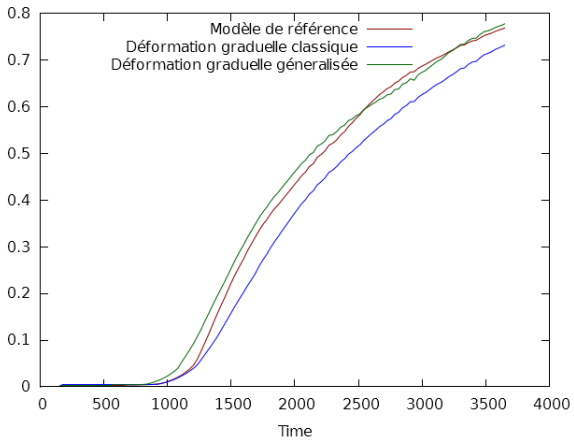
On peut remarquer qu'ils sont tous très éloignés des rayons optimaux trouvés en section précédente, l'algorithme d'optimisation est donc très probablement tombé dans un minimum local et n'a pas été capable en sortir.

3.2.3.4 Calage d'historique obtenu

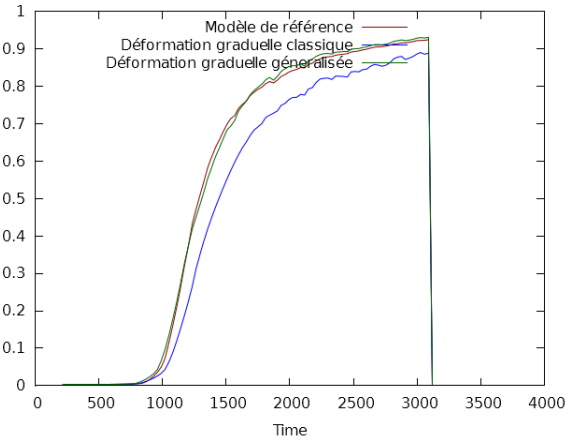
On illustre dans ce paragraphe la qualité du modèle obtenu par le meilleur calage d'historique réalisé précédemment. On donne en figure 3.15 les historiques de WC simulés avant et après optimisation pour les 3 premiers puits autour desquels des zones ont été créées (P-10, P-13 et P-15). Même si une erreur est toujours présente entre le modèle de référence et le modèle optimisé, on peut observer un net progrès dans le calage des données.



(a) P-10



(b) P-13



(c) P-15

FIGURE 3.15 – Calage du WC pour les puits P-10, P-13 et P-15, cas Brugge.

3.2.3.5 Seconde optimisation

Afin de déterminer s'il est légitime d'appliquer les méthodes de paramétrisation séquentiellement, un second calage d'historique a été réalisé en prenant en compte les résultats du premier calage obtenu avec sept paramètres de rayon initialisés à $525m$ (voir figure 3.13). On choisit pour réalisation de base la meilleure réalisation obtenue après le premier calage d'historique et on réapplique la méthodologie de patchwork avec les 20 réalisations de modèle indépendantes générées initialement. Les valeurs de la fonction objectif pour la nouvelle réalisation de base sont montrées dans le tableau 3.5.

Cette méthodologie conduit à la création de 9 zones autour des puits P-10, P-11, P-12, P-13, P-14, P-17, P-19, P-20 et P-8. Le rayon des zones a été choisi initialement à $225m$. Des tests ont été réalisés pour les trois variantes de la méthode de déformation graduelle généralisée décrites précédemment : la méthode de déformation graduelle locale (taille des zones invariantes), la méthode de déformation graduelle généralisée avec un unique rayon régissant toutes les zones et la méthode de déformation graduelle généralisée. Les résultats de ces tests sont présentés en figure 3.16.

On constate tout d'abord que la méthode de patchwork parvient encore à améliorer la fonction objectif (6.97 pour la réalisation de base contre 6.41 après patchwork). Ensuite, comme

TABLE 3.5 – Valeurs de la fonction objectif et des fonctions objectif locales pour la meilleure réalisation de modèle du cas Brugge obtenue.

F	I-1 P-1 P-19	I-2 P-10 P-2	I-3 P-11 P-20	I-4 P-12 P-3	I-5 P-13 P-4	I-6 P-14 P-5	I-7 P-15 P-6	I-8 P-16 P-7	I-9 P-17 P-8	I-10 P-18 P-9
7.18	0.8e-02 7.8e-02 4.6e-01	1.2e-03 1.0e+00 1.4e-01	1.5e-03 3.2e-01 1.8e+00	3.3e-03 2.9e-01 4.2e-02	6.7e-03 1.7e-01 1.7e-02	3.1e-03 4.0e-01 5.8e-01	2.0e-03 6.6e-02 8.2e-03	2.1e-03 9.6e-02 6.1e-02	7.2e-03 1.0e+00 1.6e-01	4.4e-03 1.3e-01 1.6e-02

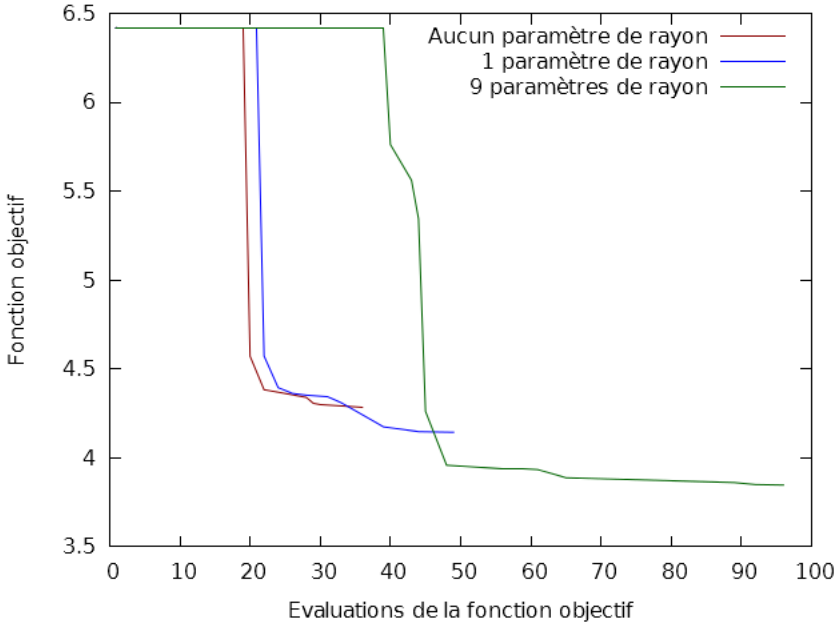


FIGURE 3.16 – Comparaison des trois variantes de la méthode de déformation graduelle généralisée sur une seconde optimisation, cas Brugge.

dans le premier cas testé, on peut remarquer que les paramétrisations avec zéro ou un paramètre de rayon donnent sensiblement les mêmes résultats (4.28 pour la méthode sans rayon contre 4.14 pour la méthode avec un rayon). Ce résultat est dû au fait que le rayon initial des zones est choisi proche de l'optimum, la possibilité de le faire varier n'apporte donc que très peu. En effet, le rayon des zones dans la meilleure réalisation est de $218m$, très proche des $225m$ choisis initialement. D'un autre côté, malgré un coût dans ce cas significativement plus élevé, introduire les rayons des zones indépendamment dans les paramètres apporte un léger gain sur la fonction objectif (3.84 pour la déformation graduelle généralisée contre 4.28 pour la méthode sans rayon, soit un gain de plus de 10%). Le tableau suivant (table 3.6) donne la valeur des rayons au point optimal.

Rayon 1	Rayon 2	Rayon 3	Rayon 4	Rayon 5	Rayon 6	Rayon 7	Rayon 8	Rayon 9
222	216	187	228	230	117	229	224	289

TABLE 3.6 – Valeurs des rayons (en mètres) après seconde optimisation de la fonction objectif.

On peut constater que sur ces 9 rayons, seuls trois se sont écartés significativement de leur valeur initiale de $225m$: Le rayon 3 ($147m$), le rayon 6 ($117m$) et le rayon 9 ($289m$). Comme

dans la section précédente, ces résultats peuvent expliquer pourquoi la paramétrisation avec un unique paramètre de rayon n'apporte qu'un très léger avantage sur la déformation graduelle locale avec zones fixes.

Ces résultats indiquent qu'il est tout à fait concevable d'envisager la réalisation d'une séquence d'optimisations utilisant la nouvelle méthodologie jusqu'à arriver à un calage des données satisfaisant. En effet, à chaque optimisation, la méthodologie de patchwork améliorée permet de créer des zones autour des puits les moins bien calés. Au cours des optimisations, chaque puits présentant une fonction objectif locale jugée non suffisamment faible se verra donc inévitablement attribuer des paramètres pour améliorer le calage de ses données. C'est ce qu'on observe dans ce deuxième calage : des zones sont créées autour de certains puits qui n'en possédaient pas au cours de la première optimisation et permettent un bien meilleur calage des données de production de ces puits. Un exemple ici est le puits producteur P-17 : la seconde optimisation a permis de faire passer sa fonction objectif locale de 1.58 à 0.16. On présente en figure 3.17 le calage des données de production du puits P-17. On voit clairement dans ce cas que la réalisation obtenue par la première optimisation calait correctement la pression en fond de puits mais pas le Water Cut, la seconde optimisation permet de remédier à ce problème.

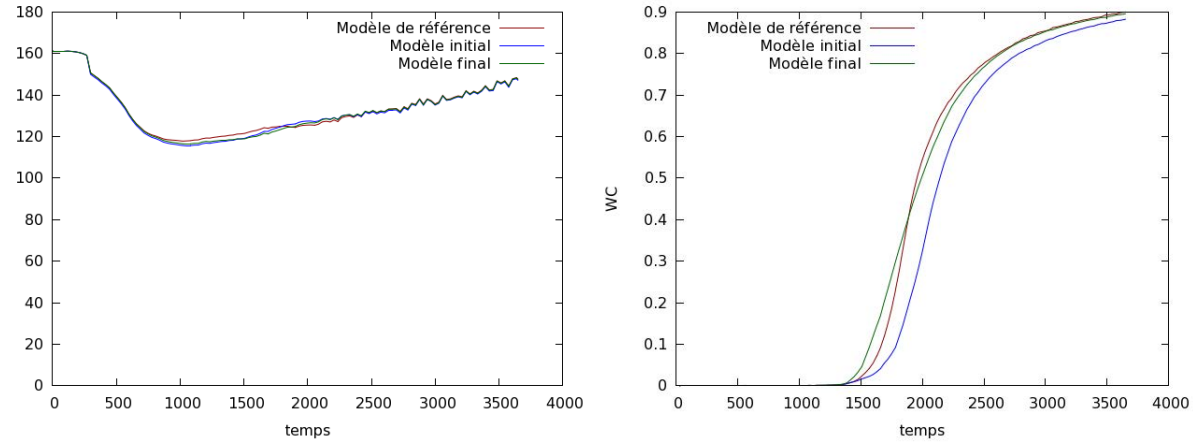


FIGURE 3.17 – Calages des données de production pour le puits producteur P-17 (BHFP à gauche et WC à droite), cas Brugge. Le modèle initial est le résultat de la première optimisation.

3.2.3.6 Discussions sur le cas Brugge

La méthodologie adoptée ici est globalement meilleure que les méthodes de paramétrisation utilisées jusqu'ici. Dans ce cas test, l'optimisation préliminaire à un paramètre donne des résultats spectaculaires, la valeur de la fonction objectif obtenue étant meilleure que celle trouvée après optimisation par déformation graduelle classique sans considération de rayon. L'ajout de rayons comme paramètres pour l'optimisation doit par contre être considérée avec soin. Si le point initial est très proche des rayons optimaux, par exemple dans notre cas particulier où presque tous les rayons optimaux sont proches du rayon initial de $525m$ (voir figure 3.13), l'ajout d'un unique paramètre de rayon ne sera pas bénéfique et il faudra considérer chaque rayon indépendamment pour affiner les calculs. En revanche, lorsque les rayons initiaux sont éloignés des rayons optimaux, l'ajout d'un seul paramètre de rayon permettra de faire décroitre correctement la fonction objectif alors que la considération de sept paramètres indépendants sera trop complexe pour l'optimiseur.

3.3 Résultats numériques d'optimisation

3.3.1 Cas PUNQ

On cherche à comparer ici les performances des méthodes d'optimisation sans dérivées SQA et DFO-PSOF sur le cas de calage d'historique du réservoir synthétique PUNQ. Le cas PUNQ ne possédant qu'un petit nombre de puits, pour s'assurer de la bonne séparation des variables, on utilisera une paramétrisation similaire à celle décrite dans les sections précédentes mais en créant systématiquement une zone autour de chaque puits. Ainsi, on combinera dans chaque zone la réalisation de base Z_0 avec la meilleure réalisation locale différente de Z_0.

3.3.1.1 Séparation des variables

Les dépendances de chaque fonction ont été sélectionnées de la façon décrite en table 3.7. On a en fait considéré que les données de production de chaque puits n'étaient influencées que par les paramètres correspondants aux puits les plus proches. La répartition correspondante des zones est montrée en figure 3.3. Dans la suite, f_P désigne la fonction objectif locale associée au puits P et X_P désigne l'ensemble des paramètres associés à P.

Fonction objectif locale	Paramètres					
f_{PRO-1}	X_{PRO-1}	X_{INJ-1}	X_{PRO-2}	X_{INJ-6}	X_{INJ-7}	
f_{INJ-1}	X_{PRO-1}	X_{INJ-1}	X_{PRO-2}	X_{INJ-2}		
f_{PRO-2}	X_{PRO-1}	X_{INJ-1}	X_{PRO-2}	X_{INJ-2}	X_{PRO-2}	X_{INJ-7}
f_{INJ-2}	X_{INJ-1}	X_{INJ-2}	X_{PRO-2}	X_{INJ-3}	X_{PRO-3}	
f_{PRO-3}	X_{INJ-2}	X_{PRO-3}	X_{INJ-3}	X_{PRO-4}	X_{INJ-4}	X_{INJ-7}
f_{INJ-3}	X_{PRO-2}	X_{INJ-3}	X_{PRO-3}			
f_{PRO-4}	X_{INJ-3}	X_{PRO-4}	X_{INJ-4}	X_{INJ-5}	X_{INJ-7}	
f_{INJ-4}	X_{PRO-2}	X_{PRO-3}	X_{INJ-4}			
f_{PRO-5}	X_{PRO-4}	X_{PRO-5}	X_{INJ-5}	X_{INJ-6}	X_{INJ-7}	
f_{INJ-5}	X_{PRO-4}	X_{PRO-5}	X_{INJ-5}	X_{INJ-6}		
f_{PRO-6}	X_{PRO-1}	X_{PRO-5}	X_{PRO-6}	X_{INJ-6}	X_{INJ-7}	
f_{INJ-6}	X_{PRO-5}	X_{PRO-6}	X_{INJ-6}			
f_{INJ-7}	X_{PRO-2}	X_{PRO-3}	X_{PRO-4}	X_{PRO-5}	X_{PRO-6}	X_{INJ-7}

TABLE 3.7 – Dépendance des fonction objectifs locales dans le cas PUNQ

On peut observer que dans le cas du réservoir PUNQ, il n'est pas évident que la fonction objectif soit partiellement séparable ni de déterminer les dépendances de chaque sous-fonction. Les tests numériques dans la suite vont donc nous permettre non seulement de montrer l'efficacité d'une méthode d'optimisation adaptée aux fonctions partiellement séparables mais aussi de valider l'hypothèse de séparabilité de la fonction objectif dans le cas PUNQ.

3.3.1.2 Tests numériques

Dans le premier test, aucun patchwork n'est réalisé pour l'instant : les calages d'historiques sont réalisés en appliquant la méthode de déformation graduelle généralisée avec une zone autour de chaque puits et en considérant la réalisation de base comme modèle initial. 13 réalisations de modèle indépendantes sont générées et un paramètre de déformation graduelle est créé autour de chaque puits, permettant de combiner dans chaque zone la meilleure réalisation locale avec la réalisation de base (les rayons sont choisis fixes dans ce test). On présente en figure 3.18 les résultats des deux calages d'historiques réalisés avec les deux algorithmes d'optimisation.

On peut observer un très net avantage de DFO-PSOF sur SQA : pour obtenir la même valeur de fonction objectif que SQA après 141 évaluations de celle-ci, DFO-PSOF n'en requiert que 48. De plus, en fin d'optimisation, la fonction objectif vaut 94.08 avec SQA contre 41.24 avec DFO-PSOF, soit un écart relatif de plus de 55%.

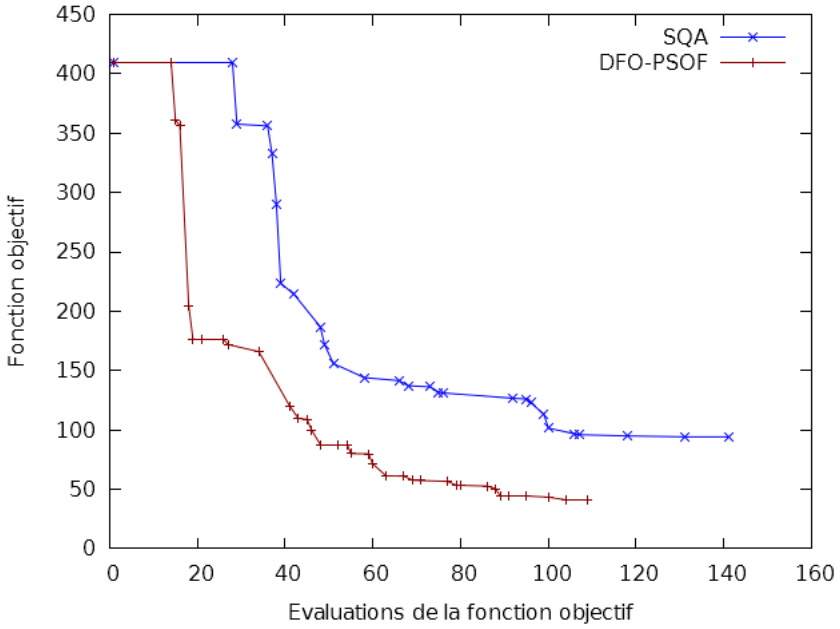


FIGURE 3.18 – Calages d'historiques réalisés avec les méthodes SQA et DFO-PSOF, cas PUNQ.

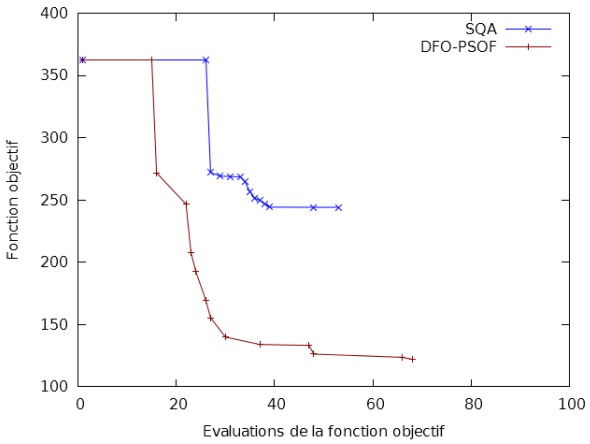
Des tests supplémentaires ont été réalisés pour comparer SQA et DFO-PSOF sur le cas PUNQ. Dans les tests suivants, on applique la méthodologie de patchwork améliorée pour déterminer la réalisation initiale et on teste les deux algorithmes d'optimisation sur les trois variantes de la méthode de déformation graduelle généralisée. On présente en figure 3.19 les résultats de ces tests.

(i) **Aucun paramètre de rayon**
- Pour obtenir la même valeur de fonction objectif que SQA en 53 évaluations de la fonction objectif, DFO-PSOF n'en requiert que 22.
- La valeur de la fonction objectif à la fin de l'optimisation est de 244.11 avec SQA et seulement de 121.99 avec DFO-PSOF, soit un gain de 50% pour la méthode DFO-PSOF.

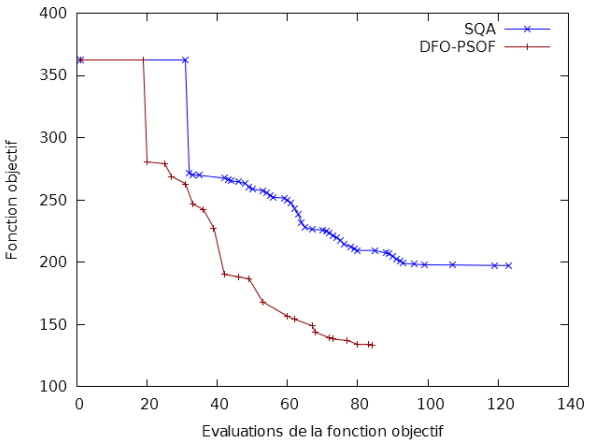
(ii) **Un paramètre de rayon**
- Pour obtenir la même valeur de fonction objectif que SQA en 123 évaluations de la fonction objectif, DFO-PSOF n'en requiert que 40.
- La valeur de la fonction objectif à la fin de l'optimisation est de 197.40 avec SQA et seulement de 133.67 avec DFO-PSOF, soit un gain de 32% pour la méthode DFO-PSOF.

(iii) **13 paramètres de rayon**
- Pour obtenir la même valeur de fonction objectif que SQA en 90 évaluations de la fonction objectif, DFO-PSOF n'en requiert que 36.
- La valeur de la fonction objectif à la fin de l'optimisation est de 215.46 avec SQA et seulement de 106.00 avec DFO-PSOF, soit un gain de 51% pour la méthode DFO-PSOF.

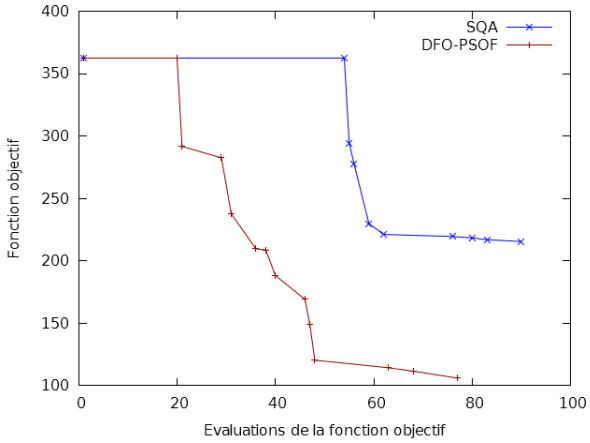
Ainsi, dans chacun des cas testés, la méthode DFO-PSOF donne de bien meilleurs résultats que SQA, ce qui confirme son intérêt pour les problèmes de calage d'historique.



(a) Aucun paramètre de rayon



(b) 1 paramètre de rayon



(c) 13 paramètres de rayon

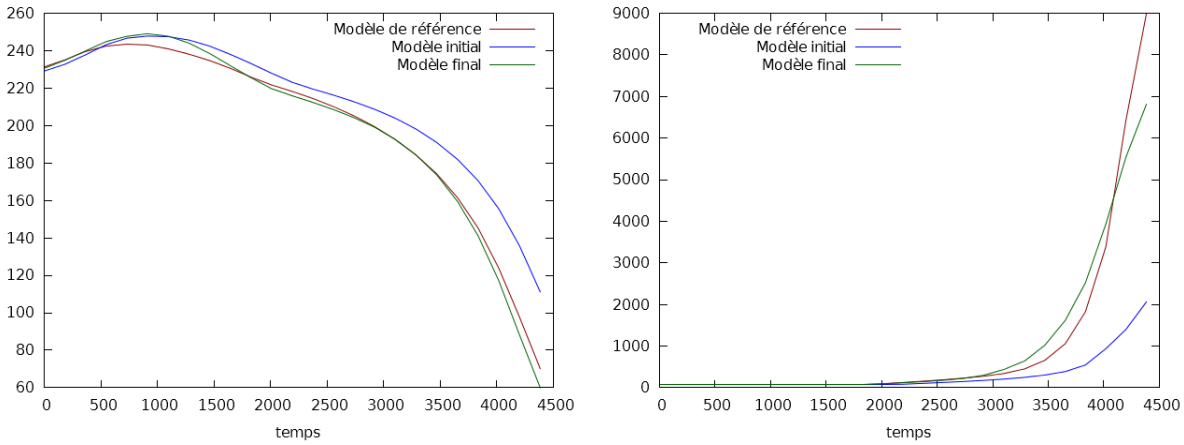
FIGURE 3.19 – Calages d'historiques réalisés avec les méthodes d'optimisation SQA et DFO-PSOF sur les trois variantes de la méthode de déformation graduelle généralisée : (a) aucun paramètre de rayon ; (b) un paramètre de rayon ; (c) 13 paramètres de rayon, cas PUNQ

3.3.1.3 Calage d'historique obtenu

On donne dans ce paragraphe quelques courbes de calage, cartes de perméabilités et cartes de saturation en huile du premier calage réalisé (figure 3.18) avec la méthode DFO-PSOF pour montrer l'intérêt de réaliser des calages d'historiques sur le cas PUNQ. La figure 3.20 montre les courbes de calage des données de production (BHFP, GOR et WC) du puits PRO-2 avec la méthode DFO-PSOF.

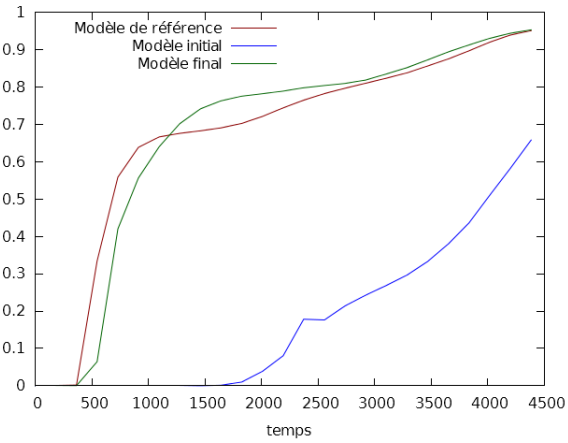
Les figures 3.21 et 3.22 reproduisent les cartes de perméabilité des 5 couches du réservoir avec à gauche le modèle de référence, au milieu le modèle initial et à droite le modèle obtenu après optimisation par DFO-PSOF. De la même façon, les figures 3.23 et 3.24 représentent les cartes de saturation en huile des cinq couches du réservoir.

Ces cartes montrent ainsi que le processus de calage d'historique permet de bien retrouver certaines caractéristiques du réservoir, notamment les zones de fortes perméabilités.



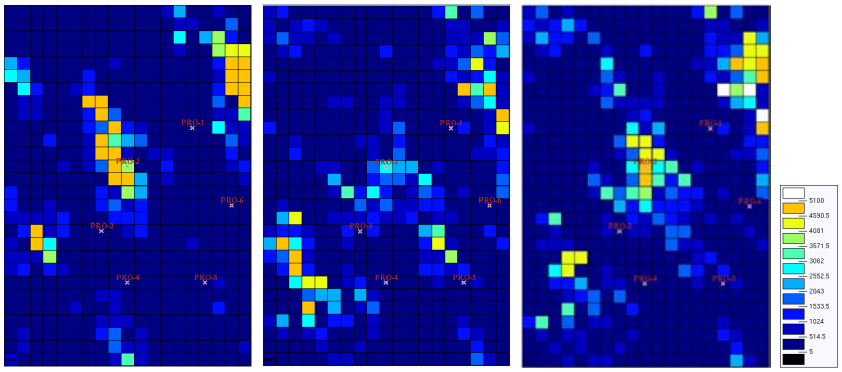
(a) BHFP

(b) GOR

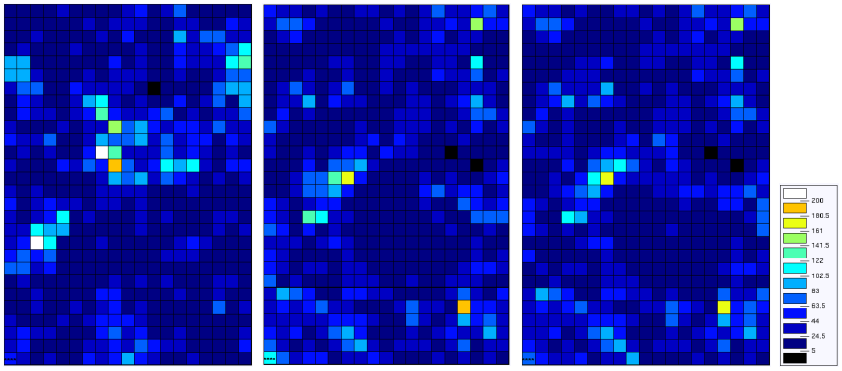


(c) WC

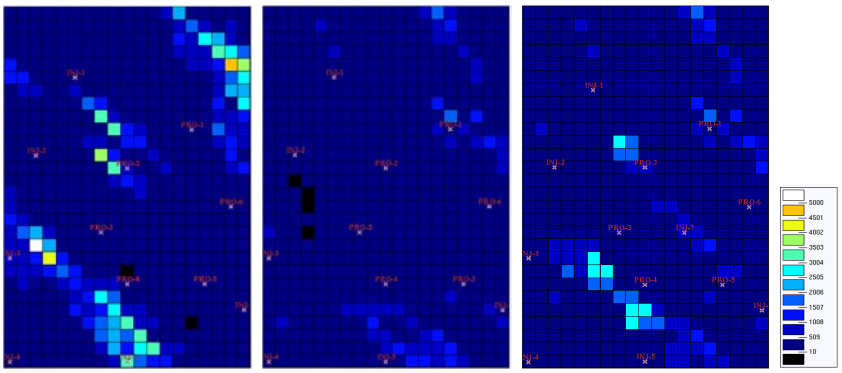
FIGURE 3.20 – Calage de l'historique de production du puits PRO-2, cas PUNQ.



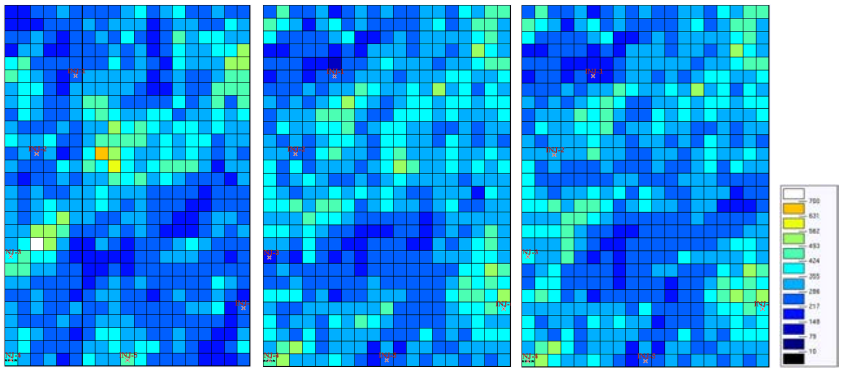
(a) couche 1



(b) couche 2

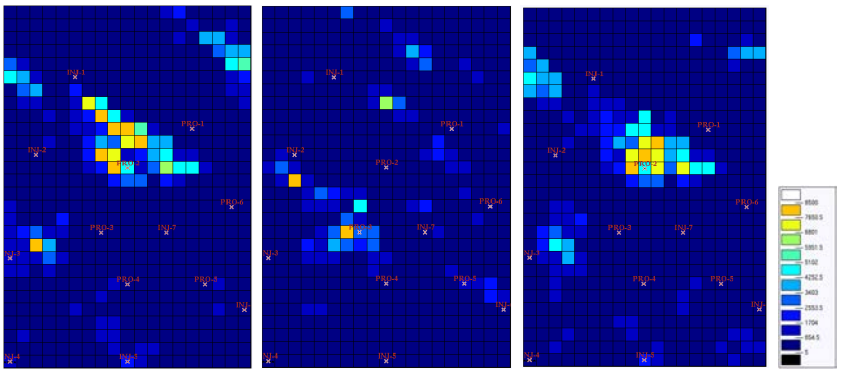


(c) couche 3



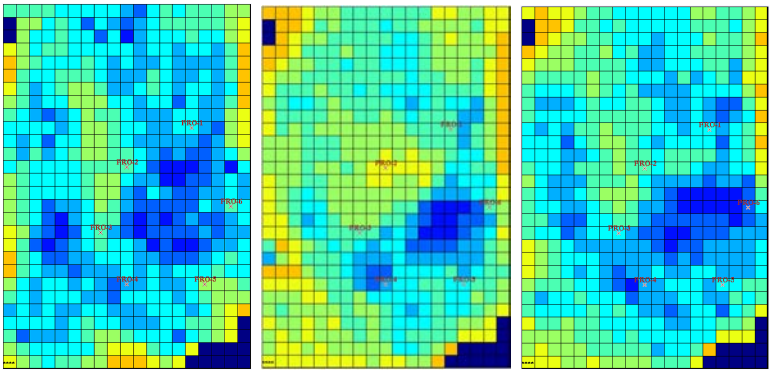
(d) couche 4

FIGURE 3.21 – Cartes des perméabilités du réservoir, cas PUNQ.

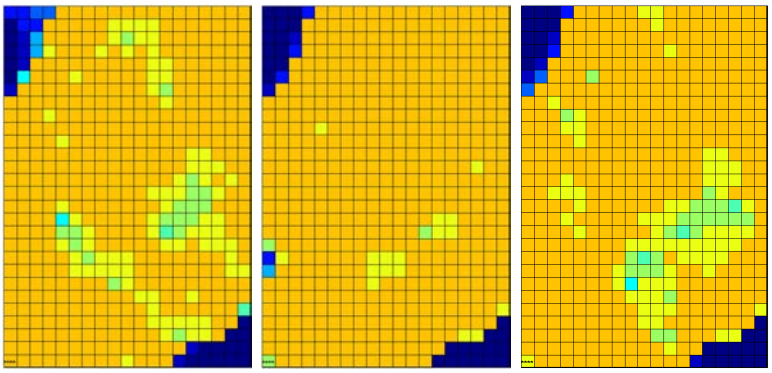


(a) couche 5

FIGURE 3.22 – Cartes des perméabilités du réservoir, cas PUNQ.

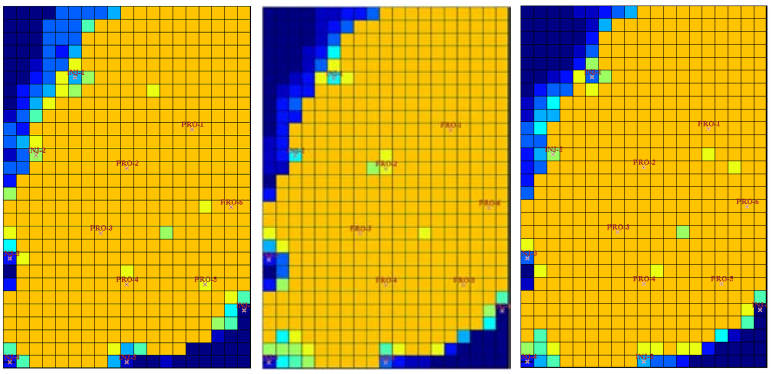


(a) couche 1

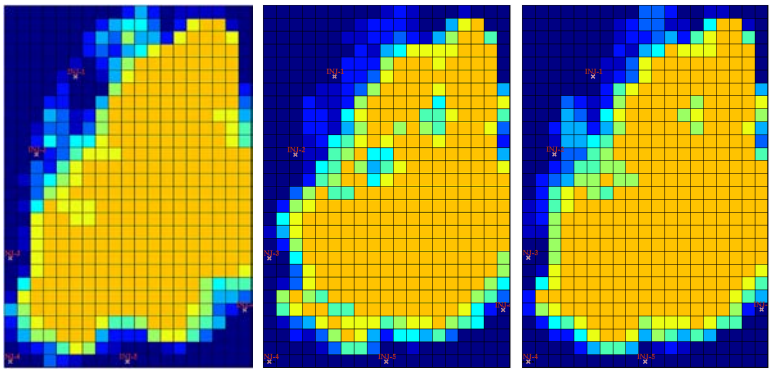


(b) couche 2

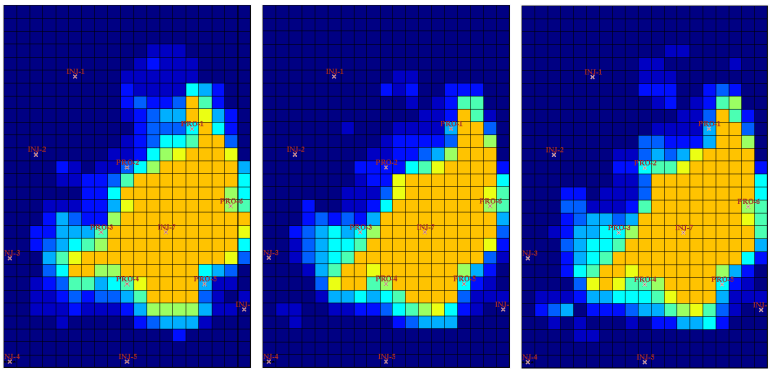
FIGURE 3.23 – Cartes des saturations en huile du réservoir, cas PUNQ.



(a) couche 3



(b) couche 4



(c) couche 5

FIGURE 3.24 – Cartes des saturations en huile du réservoir, cas PUNQ.

3.3.2 Le cas Brugge

On compare ici les méthodes d'optimisation SQA et DFO-PSOF sur le cas Brugge. On reprend la paramétrisation de la section 3.2.3, où sept zones sont créées autour des puits P-10, P-13, P-15, P-16, P-19, P-2 et P-5 et où un paramètre de déformation graduelle est introduit pour chacune de ces zones. Les trois variantes de la méthode de déformation graduelle généralisée (rayons fixes, un paramètre contrôlant tous les rayons simultanément et un paramètre pour chaque rayon de zone) sont ici testées.

3.3.2.1 Séparation des variables

Chaque puits producteur possède une fonction objectif locale mais seuls les puits autour desquels des zones ont été créées sont associés des paramètres. La figure 3.8 (section 3.2.3) représente la répartition des zones dans le réservoir, les dépendances sont alors déterminées en fonction des distances entre les puits et les zones. De plus, les données de production des puits injecteurs sont considérées indépendantes des puits producteurs. Dans la suite, f_P désigne la fonction objectif locale associée au puits P et X_P désigne l'ensemble des paramètres associés à P. Les dépendances sont précisées dans le tableau suivant (table 3.8).

Fonction objectif locale	Paramètres
f_{P-1}	X_{P-16}
f_{P-2}	X_{P-10} X_{P-13} X_{P-2}
f_{P-3}	X_{P-16} X_{P-19}
f_{P-4}	X_{P-16} X_{P-19}
f_{P-5}	X_{P-19} X_{P-5}
f_{P-6}	X_{P-19} X_{P-5}
f_{P-7}	X_{P-19}
f_{P-8}	X_{P-16}
f_{P-9}	X_{P-10} X_{P-13} X_{P-2}
f_{P-10}	X_{P-10} X_{P-13} X_{P-2}
f_{P-11}	X_{P-10} X_{P-13} X_{P-2}
f_{P-12}	X_{P-10} X_{P-13} X_{P-2}
f_{P-13}	X_{P-10} X_{P-13} X_{P-15} X_{P-2}
f_{P-14}	X_{P-13} X_{P-15} X_{P-16} X_{P-2}
f_{P-15}	X_{P-13} X_{P-15} X_{P-16}
f_{P-16}	X_{P-15} X_{P-16}
f_{P-17}	X_{P-15} X_{P-16} X_{P-19}
f_{P-18}	X_{P-19}
f_{P-19}	X_{P-19} X_{P-5}
f_{P-20}	X_{P-19} X_{P-5}

TABLE 3.8 – Dépendance des fonctions objectifs locales dans le cas Brugge

3.3.2.2 Tests numériques

Un premier test a été réalisé dans lequel aucun paramètre de rayon n'a été introduit. La figure 3.25 présente les résultats de l'optimisation. On peut constater sur cette figure que la méthode DFO-PSOF permet une décroissance bien plus rapide de la fonction objectif que la méthode SQA. En effet, la valeur de la fonction objectif est la même en fin d'optimisation pour

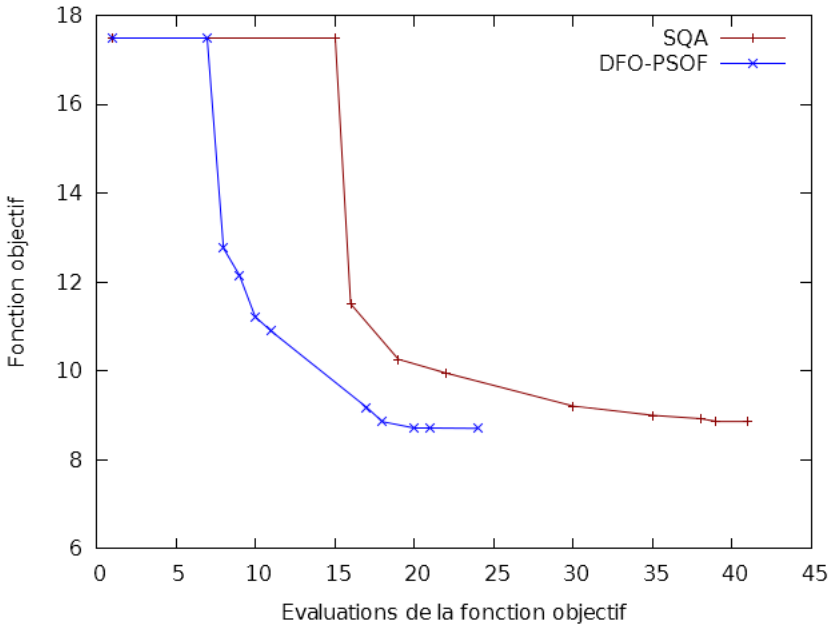


FIGURE 3.25 – Comparaison entre SQA et DFO-PSOF dans le cas où aucun paramètre de rayon n'a été introduit, cas Brugge.

les deux méthodes (8.70 pour DFO-PSOF contre 8.86 pour SQA) mais la méthode DFO-PSOF requiert deux fois moins d'évaluations pour y parvenir (24 contre 41).

Un deuxième test a été réalisé avec sept paramètres de rayon en supplément des paramètres de déformation graduelle. La figure 3.26 montre les résultats de ce test.

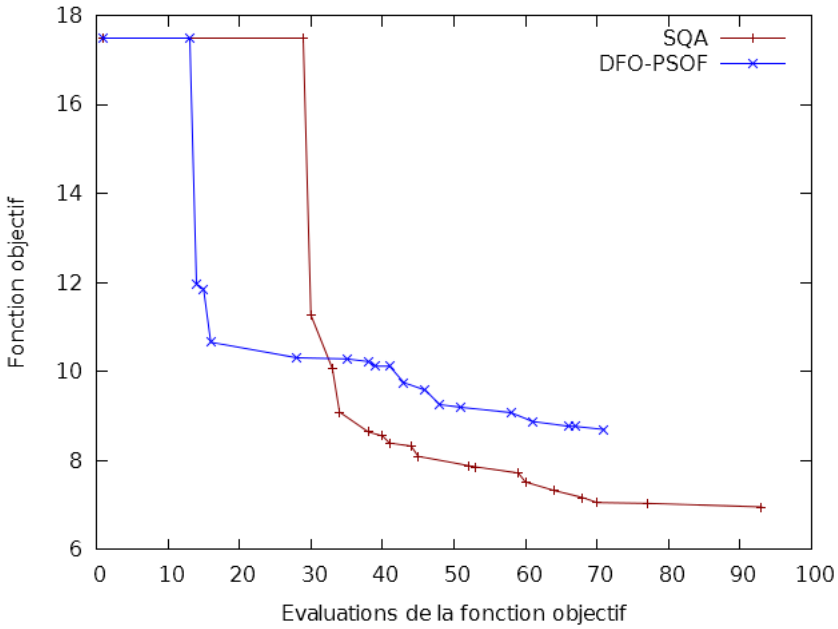


FIGURE 3.26 – Comparaison entre SQA et DFO-PSOF dans le cas où sept paramètres de rayon indépendants ont été introduits, cas Brugge.

Comme dans le cas précédent, on remarque que la décroissance de la fonction objectif est bien plus rapide pour la méthode DFO-PSOF. Néanmoins, la méthode n'arrive pas aussi loin que SQA (8.39 contre 7.18 en fin d'optimisation) et n'apporte même pas un progrès flagrant par rapport au cas sans rayon (8.39 contre 8.70). Ce résultat peut être dû à plusieurs causes, par exemple :

— l'hypothèse de séparabilité peut n'être pas aussi valide pour les paramètres de rayon dans le cas Brugge, en particulier lorsqu'on considère des rayons assez grands de l'ordre de $525m$.

— en ajoutant les paramètres de rayon, la qualité de la séparation des variables a été déteriorée (particulièrement avec des rayons aussi grands). Il est donc possible que le gain obtenu sur la qualité des modèles quadratiques en construisant un modèle pour chaque fonction objectif ne soit pas suffisant pour contrebalancer l'approximation faite avec l'hypothèse de séparabilité partielle.

Les performances des deux algorithmes d'optimisation ont ensuite été testée lorsque les rayons sont arbitrairement fixés à $225m$ afin de déterminer l'influence de la taille des rayons sur la qualité de la séparation des variables. On ne teste dans ce paragraphe que les cas sans paramètre de rayon et avec un unique paramètre régissant la taille de toutes les zones, le cas avec 7 paramètres ayant donné des résultats décevants (voir section 3.2.3).

On commence par présenter en figure 3.27 les résultats sans introduire le rayon des zones.

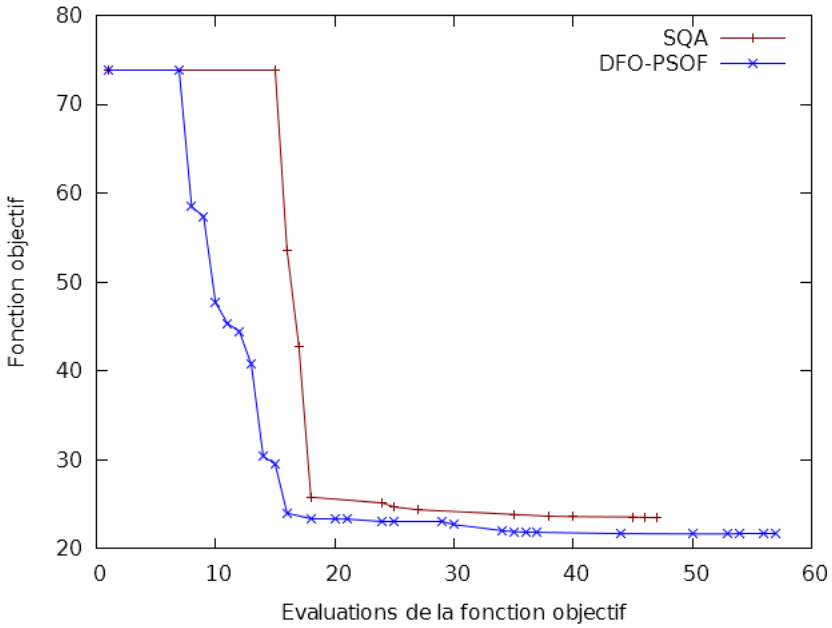


FIGURE 3.27 – Comparaison entre SQA et DFO-PSOF dans le cas où aucun paramètre de rayon n'a été introduit, cas Brugge.

De la même façon que dans le test précédent (figure 3.25), la décroissance de la fonction objectif est plus rapide avec la méthode DFO-PSOF en présentant une valeur finale légèrement meilleure (21.65 pour la méthode DFO-PSOF contre 23.50 pour SQA, soit un gain relatif d'environ 7%).

Le cas avec un unique paramètre de rayon est quant à lui présenté en figure 3.28.

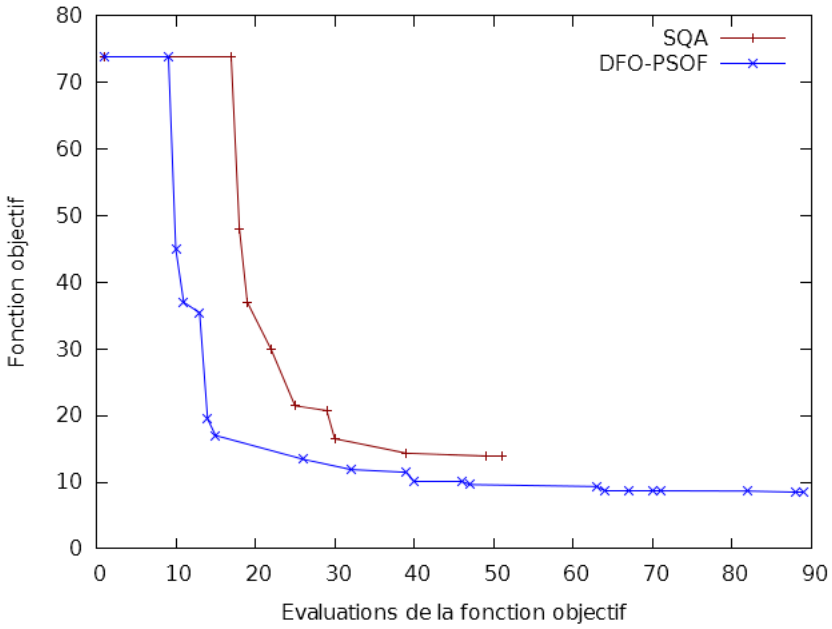


FIGURE 3.28 – Comparaison entre SQA et DFO-PSOF dans le cas où un unique paramètre de rayon a été introduit, cas Brugge.

Ici encore la décroissance de la fonction objectif est bien plus rapide pour la méthode DFO-PSOF. Le résultat en fin d'optimisation est dans ce cas bien meilleur que pour la méthode SQA (8.58 contre 13.99 soit un gain relatif de plus de 33%). Il est comparable au résultat obtenu avec le meilleur point initial avec la méthode SQA en un nombre similaire d'évaluations (rayons choisis à $525m$, voir figure 3.13). Lorsqu'on introduit un paramètre de rayon unique, on considère qu'il est influent dans toutes les fonctions objectif locales. Ce simple fait implique que la méthode trouve très vite le rayon optimal et cherche à affiner les paramètres de déformation graduelle ensuite. Dans ce test, le paramètre de rayon varie beaucoup dans les premières itérations jusqu'à parvenir à $525m$ en environ 20 simulations pour ne changer que marginalement dans les simulations suivantes.

3.3.2.3 Seconde optimisation

Pour tester la validité de l'hypothèse de séparation des variables dans le cas où la fonction objectif est plus faible, on compare les performances des algorithmes SQA et DFO-PSOF sur le second calage présenté en section 3.2.3.5. Le choix de séparation des variables est donné dans la table 3.9. La figure 3.29 compare les résultats des algorithmes SQA et DFO-PSOF avec cette décomposition de la fonction objectif. Pour les trois cas suivants :

(i) **aucun paramètre de rayon**
- Pour obtenir la même valeur de fonction objectif que SQA après 36 évaluations, DFO-PSOF n'en requiert que 17.
- La valeur de la fonction objectif à la fin de l'optimisation est de 4.22 avec SQA et seulement de 3.87 avec DFO-PSOF, soit un gain de 8% pour la méthode DFO-PSOF.

(ii) **un paramètre de rayon**
- Pour obtenir la même valeur de fonction objectif que SQA après 49 évaluations, DFO-PSOF n'en requiert que 28.

Fonction objectif locale	Paramètres			
f_{P-1}	X_{P-14}	X_{P-8}		
f_{P-2}	X_{P-10}	X_{P-11}	X_{P-12}	X_{P-13}
f_{P-3}	X_{P-17}	X_{P-8}		
f_{P-4}	X_{P-17}	X_{P-8}		
f_{P-5}	X_{P-20}			
f_{P-6}	X_{P-19}	X_{P-20}		
f_{P-7}	X_{P-17}	X_{P-19}	X_{P-8}	
f_{P-8}	X_{P-17}	X_{P-8}		
f_{P-9}	X_{P-10}	X_{P-13}		
f_{P-10}	X_{P-10}	X_{P-11}	X_{P-12}	
f_{P-11}	X_{P-10}	X_{P-11}	X_{P-12}	
f_{P-12}	X_{P-10}	X_{P-11}	X_{P-12}	X_{P-13}
f_{P-13}	X_{P-12}	X_{P-13}	X_{P-14}	
f_{P-14}	X_{P-13}	X_{P-14}		
f_{P-15}	X_{P-14}	X_{P-17}		
f_{P-16}	X_{P-17}	X_{P-8}		
f_{P-17}	X_{P-17}	X_{P-8}		
f_{P-18}	X_{P-17}	X_{P-19}		
f_{P-19}	X_{P-19}	X_{P-20}		
f_{P-20}	X_{P-19}	X_{P-20}		

TABLE 3.9 – Dépendance des fonctions objectif locales dans le cas Brugge (seconde optimisation).

— La valeur de la fonction objectif à la fin de l'optimisation est de 4.14 avec SQA et seulement de 3.93 avec DFO-PSOF.

(iii) **9 paramètres de rayon**

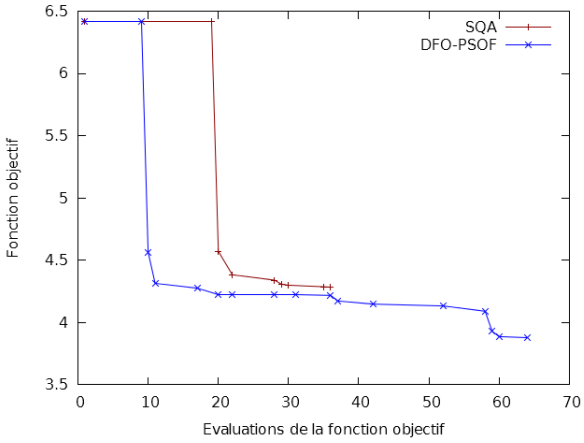
— Pour obtenir la même valeur de fonction objectif que SQA après 96 évaluations, DFO-PSOF n'en requiert que 71.

— La valeur de la fonction objectif à la fin de l'optimisation est de 3.84 avec SQA et seulement de 3.56 avec DFO-PSOF.

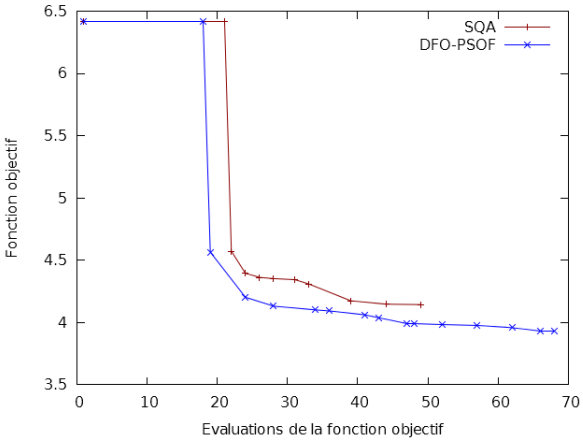
Dans les trois cas testés, les deux méthodes donnent un point final similaire mais DFO-PSOF offre une vitesse de convergence plus importante. Ces bons résultats nous amènent à penser que l'hypothèse de séparabilité partielle de la fonction objectif peut rester valable même lorsqu'on cherche à affiner le modèle.

3.3.2.4 Calage d'historique obtenu

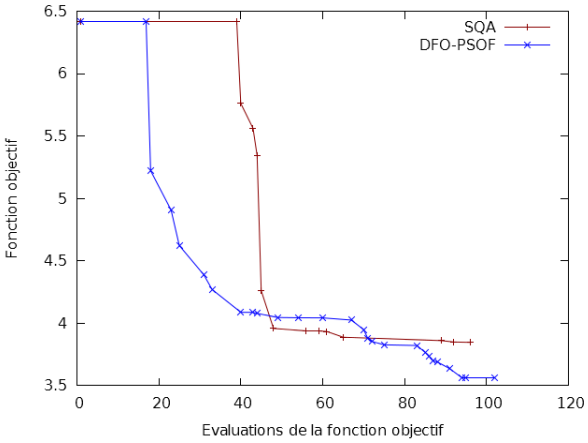
On donne dans ce paragraphe quelques courbes de calage, cartes de perméabilités et cartes de saturation en huile du meilleur calage effectué pour montrer l'intérêt de réaliser des calages d'historiques sur le cas Brugge. La fonction objectif étant très faible pour la meilleure réalisation de modèle obtenue, les courbes de production sont très proches des courbes de référence. La figure 3.30 montre le calage des historiques de WC et BHFP pour un puits moyen (puits producteur P-2, fonction objectif locale de 0.16). La figure 3.31 quant à elle montre les calages des historiques de WC et BHFP pour le puits producteur P-10 présentant la plus mauvaise fonction objectif locale (1.24). Enfin, la figure 3.32 donne les courbes de calage pour le puits producteur P-6 présentant la meilleure fonction objectif locale (0.00227). Les historiques de WC du puits P-6 semblent ne pas donner de bons résultats, cependant on peut remarquer que la quantité d'eau



(a) Aucun paramètre de rayon



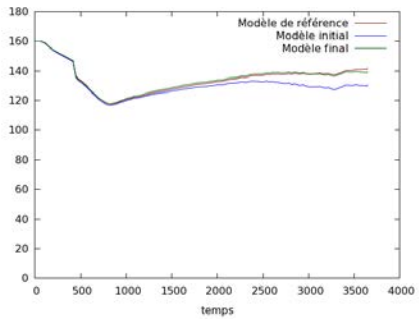
(b) 1 paramètre de rayon



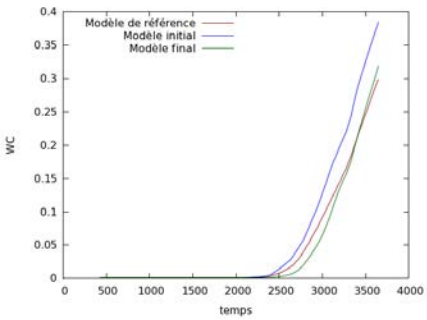
(c) 9 paramètres de rayon

FIGURE 3.29 – Comparaison des algorithmes SQA et DFO-PSOF sur le deuxième calage du cas Brugge.

produite par ce puits est très faible comparée aux autres puits. L'erreur commise dans la fonction objectif est donc très faible malgré l'écart graphique apparent entre les deux courbes.

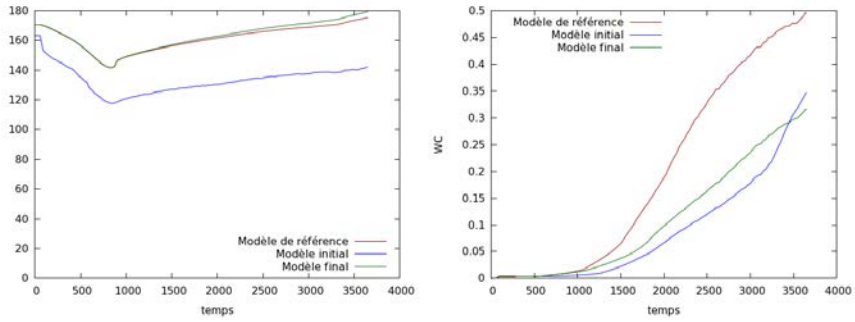


(a) BHFP



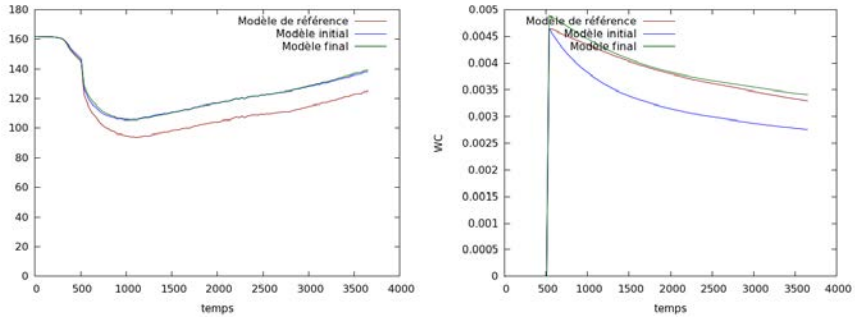
(b) WC

FIGURE 3.30 – Calages des données de production pour le puits producteur P-2, cas Brugge.



(a) BHFP (b) WC

FIGURE 3.31 – Calages des données de production pour le puits producteur P-10, cas Brugge.

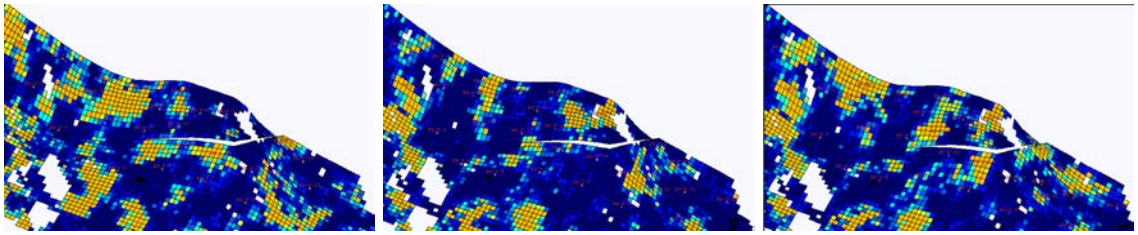


(a) BHFP (b) WC

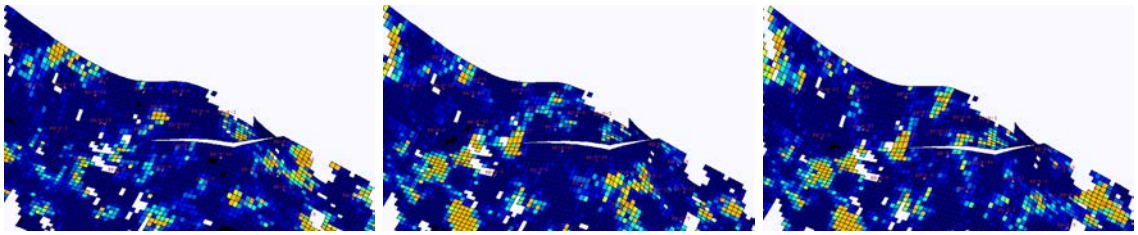
FIGURE 3.32 – Calages des données de production pour le puits producteur P-6, cas Brugge.

On présente les cartes de perméabilité et de saturation en huile de la meilleure réalisation obtenue en figure 3.33. Elle montre les cartes de perméabilité des 5 couches à forte perméabilité du réservoir (couches 1, 2, 6, 7 et 8). Pour chacune des couches, on présente la réalisation de référence à gauche, la réalisation de base au milieu et la réalisation obtenue après calage d'historique à droite. À nouveau, on peut remarquer que même si la réalisation optimale obtenue ne retrouve pas parfaitement les cartes de perméabilité de la réalisation de référence, elle est malgré tout bien plus proche autour des puits ayant suscité la création de zones.

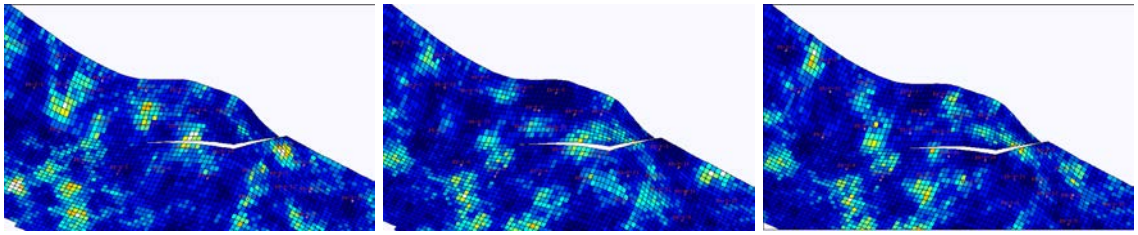
La figure 3.34 montre enfin les cartes de saturation en huile de la première couche du réservoir. La première ligne donne les cartes au temps initial pour respectivement les cas de référence (à gauche), initial (au milieu) et final (à droite) tandis que la deuxième ligne donne les mêmes cartes au temps final. Ici encore, le calage d'historique permet de mieux approcher le comportement du réservoir.



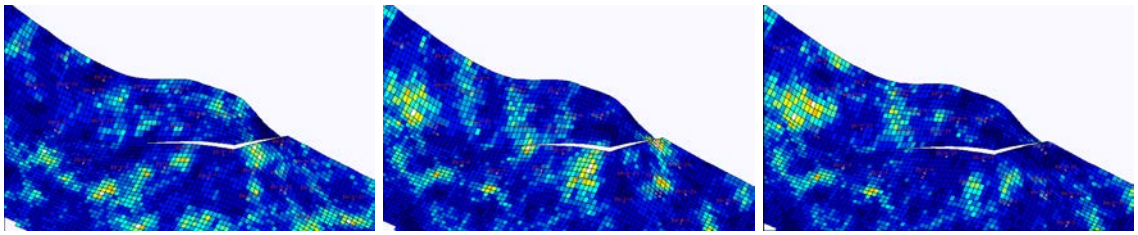
(a) couche 1



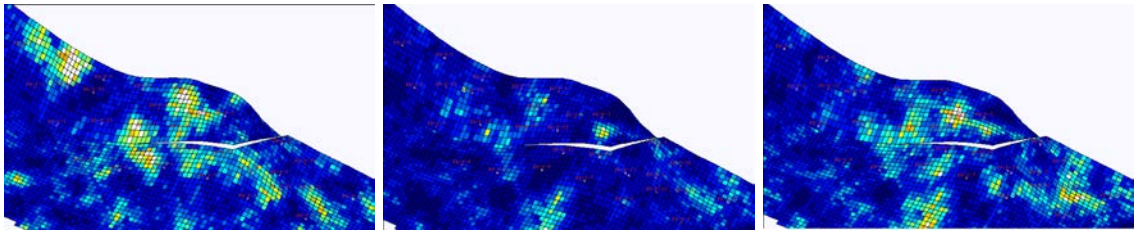
(b) couche 2



(c) couche 6

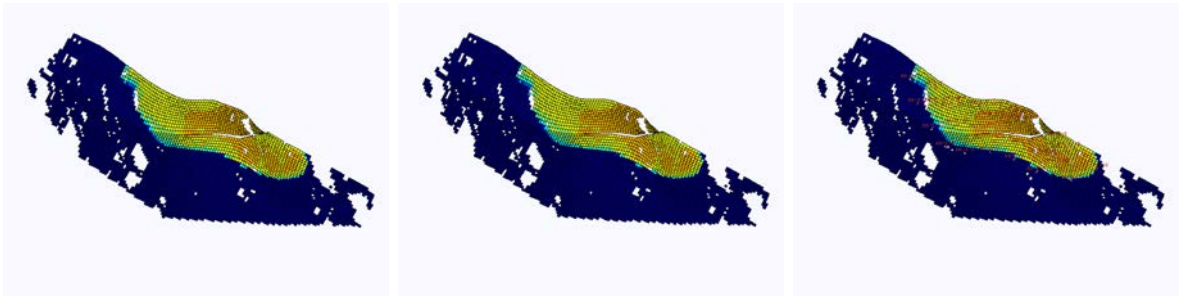


(d) couche 7

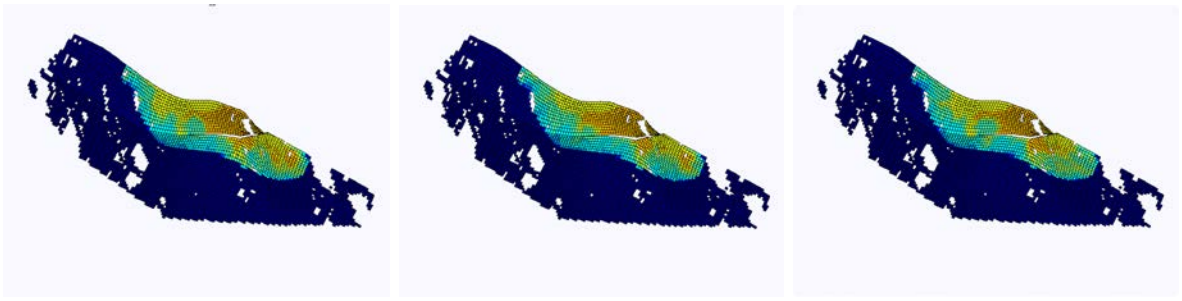


(e) couche 8

FIGURE 3.33 – Cartes de perméabilité pour les réalisations de référence (à gauche), de base (au milieu) et optimale (à droite),cas Brugge.



(a) temps initial



(b) temps final

FIGURE 3.34 – Cartes de saturaton en huile de la couche superficielle du réservoir pour les réalisations de référence (à gauche), de base (au milieu) et optimale (à droite), cas Brugge.

3.4 Bilan du chapitre

La méthode de déformation graduelle généralisée ainsi que la méthode d'optimisation DFO-PSOF ont été testées sur les deux cas tests classiques en ingénierie de réservoir, en l'occurrence les cas PUNQ et Brugge.

Les tests la nouvelle méthode de paramétrisation montrent qu'elle donne dans la presque totalité des cas de meilleurs résultats que les méthodes existantes. Laisser la possibilité aux zones de changer de taille au cours de l'optimisation apporte toujours de larges gains dans la diminution de la fonction objectif. La difficulté réside alors dans le choix de considérer ou non les formes des zones indépendamment. En effet, dans certains cas, considérer chaque rayon de zone indépendamment n'apporte pas de gain et ne fait qu'augmenter le coût de calcul tandis que dans d'autres, ne considérer qu'un seul rayon pour toutes les zones peut ne pas être suffisant pour présenter une amélioration.

La méthode d'optimisation DFO-PSOF donne elle aussi de très bons résultats sur les deux cas tests. Ceci confirme en particulier la validité de l'hypothèse de séparabilité partielle sur des cas industriels lorsque les paramètres sont bien choisis ainsi que la rapidité de convergence de cette méthode dans de tels cas complexes.

Conclusion

Nous avons tenté dans cette thèse d'apporter des solutions à deux aspects distincts du problème inverse de calage d'historique rencontré en ingénierie pétrolière :

(i) la définition de paramètres pertinents pour le problème,

(ii) le développement d'une méthode d'optimisation sans dérivées n'utilisant qu'un nombre restreint de simulations d'écoulements pour arriver à convergence.

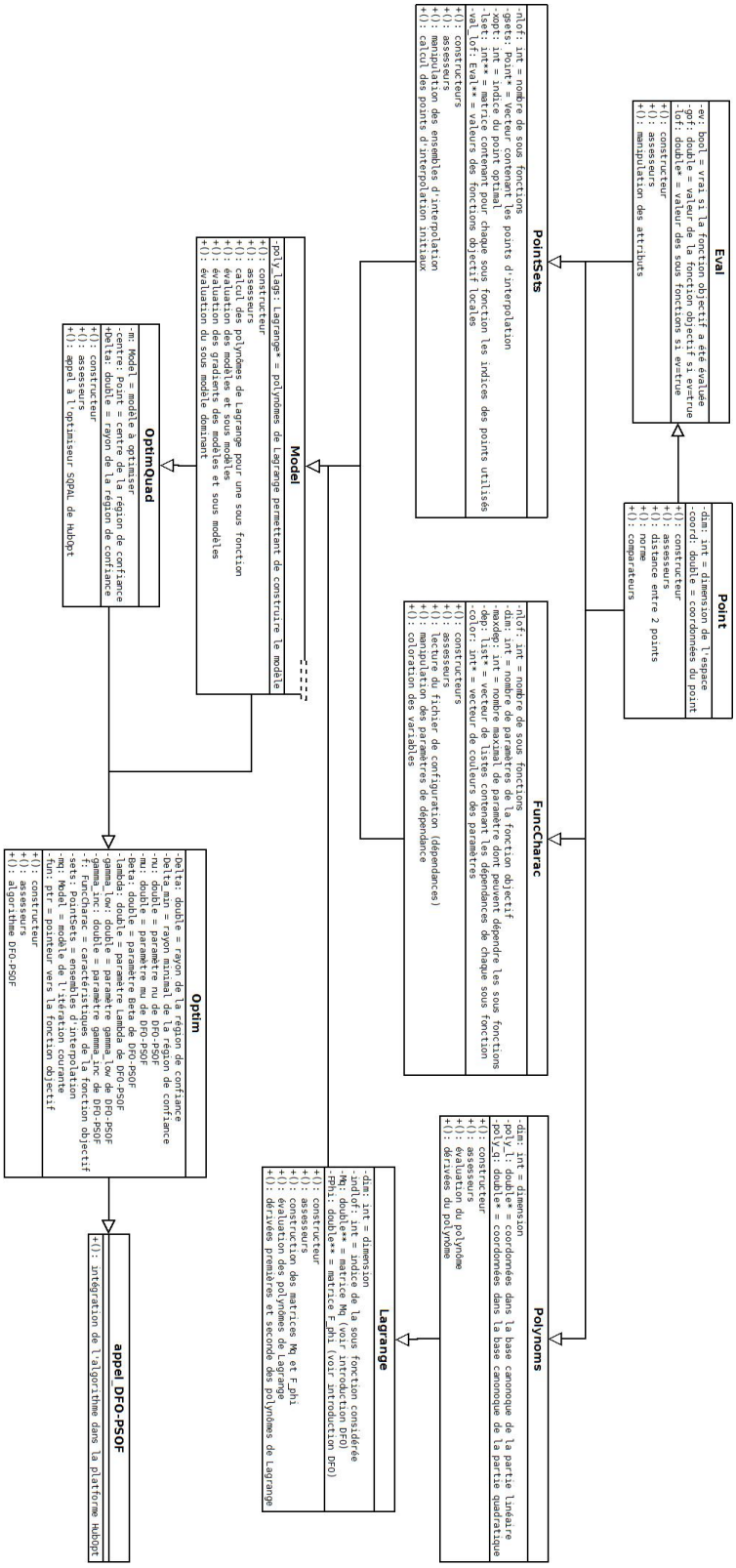
Grâce à une hypothèse de séparabilité partielle de la fonction objectif, une nouvelle méthode de paramétrisation géostatique ainsi qu'un nouvel algorithme d'optimisation sans dérivées adapté au problème, nommé DFO-PSOF, ont été présentés.

Dans un premier temps, la méthode classique de déformation graduelle locale a été généralisée de façon à ce qu'elle soit moins dépendante du découpage du réservoir initial. Afin d'y parvenir, nous avons fait en sorte d'être capables de modifier dynamiquement au cours de l'optimisation la forme des domaines dans lesquels sont effectuées les déformations graduelles. Ainsi, même avec un mauvais choix de zones initiales, le processus d'optimisation sera capable d'ajuster leurs formes pour obtenir un meilleur calage des données de production final. De plus, cette nouvelle méthode de paramétrisation permet d'identifier simplement les indépendances entre paramètres et données de production, et donc de déterminer la séparabilité partielle de la fonction objectif. Des tests numériques sur plusieurs cas réservoir ont montré que la prise en compte de la forme des domaines permet d'améliorer significativement la décroissance de la fonction objectif sans augmenter le coût de son optimisation.

Dans une autre partie du travail, un algorithme sans dérivées de type région de confiance a été développé lorsque celle-ci présente une séparabilité partielle dans le but de limiter au maximum le nombre d'évaluations de la fonction objectif requis. Les premiers tests numériques de l'algorithme sur des fonctions analytiques connues ont même montré que le nombre total d'évaluations requises ne dépendait pas du nombre de paramètres mais seulement de la bonne séparation de la fonction objectif. Une preuve théorique de la convergence de l'algorithme a par ailleurs été apportée. Les très bons résultats obtenus sur plusieurs cas de réservoir ont enfin permis de valider à la fois l'hypothèse de séparabilité partielle de la fonction objectif et la pertinence de l'utilisation de ce nouvel algorithme pour résoudre des problèmes de calage d'historique.

En résumé, une nouvelle méthodologie complète pour le calage d'historique est proposée ici, depuis la paramétrisation du problème jusqu'à son optimisation, conduisant à des gains numériques significatifs dans chaque étape comparativement aux méthodes existantes.

Annexe : diagramme de classe du code DFO-PSOF



Bibliographie

[1] A. Alhuthali, A. Oyerinde, and A. Datta-Gupta. Optimal waterflood management using rate control. *SPE Saudia Arabia Section Technical Symposium*, 10(05) :539–551, 2007.

[2] C. Audet and J. E. Jr Dennis. Analysis of generalized pattern search. *SIAM Journal on Optimization*, 13(3) :889–903, 2003.

[3] C. Audet, J.E. Dennis, and S. Le Digabel. Globalization strategies for mesh adaptive direct search. *Computational Optimization and Applications*, 46(2) :193–215, 2010.

[4] W. Bangerth, H. Klie, V. Matossian, M. Parashar, and M.F. Wheeler. An autonomic reservoir framework for the stochastic optimization of well placement. *Cluster Computing*, 8(4) :255–269, 2005.

[5] W. Bangerth, H. Klie, M.F. Wheeler, P.L. Stoffa, and M.K. Sen. On optimization algorithms for the reservoir oil well placement problem. *Computational Geosciences*, 2006.

[6] J.W. Barker, M. Cuypers, and L. Holden. Quantifying uncertainty in production forecasts : Another look at the PUNQ-S3 problem. *SPE Journal*, 6(4) :433–441, 2000.

[7] B.L. Beckner and X. Song. Field development planning using simulated annealing-optimal economic well scheduling and placement. *SPE annual technical conference and exhibition*, 1995.

[8] M.C. Bellout, C.D. Echeverréa, L.J. Durlofsky, B. Foss, and J. Kleppe. Joint optimization of oil well placement and controls. *Computational Geosciences*, 16(4) :1061–1079, 2012.

[9] F.V. Berghen and H. Bersini. Performance of condor, a parallel, constrained extension of powell's uobyqa algorithm : experimental results and comparison with the dfo algorithm. *Journal of Computational and Applied Mathematics*, 181 :157–175, 2005.

[10] R.C. Bissel. Calculating optimal parameters for history matching. *4th European Conference on the Mathematics of Oil Recovery*, 2, 1994.

[11] R.C. Bissell, Y. Sharma, and J.E. Killough. History matching using the method of gradients : two case studies. *SPE Annual Technical Conference and Exhibition*, 1994.

[12] A. Björk. *Numerical methods for least squares problems.* SIAM Publications, 1996.

[13] Z. Bouzarkouna. *Well placement optimization.* PhD thesis, Université Paris-Sud 11, 2012.

[14] C.G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6 :76–90, 1970.

[15] R.H. Byrd, R.B. Schnabel, and G.A. Schultz. A trust region algorithm for non linearly constrained optimization. *SIAM Journal on Numerical Analysis*, 24 :1152–1170, 1987.

[16] M.R. Cellis, J.E. Dennis, and R.A. Tapia. A trust region strategy for nonlinear equality constrained optimization. *P.T. Boggs, R.H. Byrd and R.B. Schnabel, eds.Numerical Optimization (SIAM, Philadelphia)*, pages 71–82, 1985.

[17] G. Chavent, M. Dupuy, and P. Lemonnier. History matching by use of optimal control theory. *SPEJ*, 15(1) :74–86, 1975.

[18] C. Chen, G. Gao, D. Hohl, L. Jin, J.C. Vink, and D. Weber. Assisted history matching using three derivative-free optimization algorithms. *SPE International*, 2012.

[19] C. Chen, Y. Wang, G. Li, and A.C. Reynolds. Closed-loop reservoir management on the Brugge test case. *Computational Geosciences*, 14(4) :691–703, 2010.

[20] D. Ciaurri, T. Mukerji, and L.J. Durlofsky. *Derivative-Free Optimization for Oil Field Operations*, volume 359. Springer Berlin Heidelberg, 2011.

[21] D.E. Ciaurri, O.J. Isebor, and L.J. Durlofsky. Application of derivative-free methodologies to generally constrained oil production optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 2(2) :134–161, 2011.

[22] T.F. Coleman and J.J. Moré. Estimation of sparse jacobian matrices and graph coloring problems. *Journal of Institute of Mathemetics and Applications*, 3 :117–119, 1983.

[23] B. Colson and P.L. Toint. Optimizing partially separable functions without derivatives. *Optimization Methods and Software*, 20(4-5) :493–508, 2005.

[24] A. R. Conn, K. Scheinberg, and L.N. Vicente. Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points. *SIAM Journal on Optimization*, 20(1) :387–415, 2009.

[25] A.R Conn, N.I.M Gould, and Ph.L. Toint. *Trust-Region Methods*. SIAM, 2000.

[26] A.R. Conn, K. Scheinberg, and L.N. Vicente. Geometry of interpolation sets in derivative free optimization. *Mathematical Programming*, 111(1-2), 2008.

[27] A.R Conn, K. Scheinberg, and L.N. Vicente. *Introduction to derivative free optimization*. MPS-SIAM Series on optimization, 2009.

[28] A.R. Curtis, M.J.D. Powell, and J.K. Reid. On the estimation of sparse jacobian matrices. *Journal of Institute of Mathemetics and Applications*, 3 :117–119, 1974.

[29] W.C. Davinson. Variable metric method for minimization. *SIAM Journal on Optimization*, 1 :1–17, 1991.

[30] C. Davis. Theory of positive linear dependence. *American Journal of Mathematics*, 76(4) :733–746, 1954.

[31] V. Dehdari and D.S. Oliver. Sequential quadratic programming for solving constrained production optimization–case study from Brugge field. *SPE journal*, 17(03) :874–884, 2012.

[32] V. Dehdari, D.S. Oliver, and C.V. Deutsch. Comparison of optimization algorithms for reservoir management with constraints - a case study. *Journal of Petroleum Science and Engineering*, 100 :41–49, 2012.

[33] F. Delbos and J.Ch. Gilbert. Global linear convergence of an augmented lagrangian algorithm for solving convex quadratic optimization problems. *Journal of Convex Analysis*, 12 :45–69, 2005.

[34] F. Delbos, J.Ch. Gilbert, R. Glowinski, and D. Sinoquet. Constrained optimization in seismic reflection tomography : a gauss-newton augmented lagrangian approach. *Geophysical Journal International*, 164 :670–684, 2006.

[35] C.V. Deutsch. *Geostatistical reservoir modeling*. Oxford University Press, 2002.

[36] D.Y. Ding and F. McKee. Using partial separability of the objective function for gradient-based optimizations in history matching. *SPE Reservoir Simulation Symposium*, 2011.

[37] D.Y. Ding and F. Roggero. History matching geostatistical model realizations using a geometrical domain based parameterization technique. *Mathematical Geosciences*, 42(4) :413–432, 2010.

[38] Sy Thanh Do. *Application of spsa-type algorithms to production optimization.* PhD thesis, University of Tulsa, 2012.

[39] G. Fasano, J.L Morales, and J. Nocedal. On the geometry phase in model-based algorithms for derivative-free optimization. *Optimization Methods and Softwares*, 24(1) :145–154, 2009.

[40] Y. Fischer, B. Marteau, and Y. Privat. Some inverse problems around the tokamak tore supra. *Communications on Pure and Applied Mathematics*, 11(6) :2327–2349, 2012.

[41] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13(3) :317–322, 1970.

[42] L. Frimannslund and T. Steihaug. Using partial separability of functions in generating set search methods for unconstrained optimisation, 2006.

[43] G. Gao, G. Li, and A.C. Reynolds. A stochastic optimization algorithm for automatic history matching. *SPE journal*, 12(2) :196–208, 2007.

[44] G. Gao and A. C. Reynolds. An improved implementation of the LBFGS algorithm for automatic history matching. *SPE Journal*, 11(1) :5–17, 2006.

[45] G. Gavalas, P. Shah, and J. Seinfeld. Reservoir history matching by bayesian estimation. *SPE Journal*, 16(6) :337–350, 1976.

[46] V. Gervais, Y. Gautier, F. Le Ravalec, and F. Roggero. History matching using local gradual deformation. *SPE EUROPEC/EAGE Conference and Exhibition*, 2007.

[47] D. Goldfarb. A family of variable metric updates derived by variational means. *Mathematics of Computation*, 24(109) :23–26, 1970.

[48] S. Gomez, O. Gosselin, and J.W. Barker. Gradient-based history matching with a global optimization method. *SPE Journal*, 6(6) :200–208, 2001.

[49] O. Gosselin, A. Cominelli, S. van den Berg, and S.D. Chowdhury. A gradient-based approach for history matching of both production and 4d seismic data. *Proceeding 7th European Conference on the Math. of Oil Recovery, Baveno, Italy*, 2000.

[50] N.I.M. Gould, D. Orban, and P.L. Toint. Numerical method for large-scale nonlinear optimization. *Acta Numerica*, 14 :299–361, 2005.

[51] M. Handels, M.J.Zandvliet, D.R. Brouwer, and J.D.Jansen. Adjoint-based well-placement optimization under production constraints. *SPE Reservoir Simulation Symposium 105797*, 2007.

[52] B.T. Hoffman and J. Caers. Regional probability perturbations for history matching. *SPE Annual Technical Conference and Exhibition*, 2003.

[53] L.Y. Hu. Combination of dependant realizations within the gradual deformation method. *Mathematical Geology*, 34(8) :953–963, 2002.

[54] L.Y. Hu, G. Blanc, and B. Noetinger. Gradual deformation and iterative calibration of sequential stochastic simulations. *Mathematical Geology*, 33(4) :475–489, 2001.

[55] P. Jacquard and C. Jain. Permeability distribution from field pressure data. *SPE journal*, 5(4) :281–294, 1965.

[56] B. Jafarpour and D.B. Mclaughlin. Reservoir characterization with the discrete cosine transform. *SPE Journal*, 14(1) :182–201, 2009.

[57] Y. Jarny, M.N. Ozisik, and J.P. Bardon. A general optimization method using adjoint equation for solving multidimensional inverse heat conduction. *International Journal of Heat and Mass Transfer*, 34(11) :2911–2919, 1991.

[58] T.R. Jensen and B. Toft. *Graph Coloring Problems*. John Wiley & Sons, 1995.

[59] M.P. Khaleta, R.G. Hanea, A.W. Heemink, and J-D Jansen. Model-reduced gradient-based history matching. *Computational Geosciences*, 15(1) :135–153, 2011.

[60] T. Kolda, R. Lewis, and V. Torczon. Optimization by direct search : New perspectives on some classical and modern methods. *SIAM Review*, 45(3) :385–482, 2003.

[61] J.C. Lagarias, J.A. Reeds, M.H.Wright, and P.E. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9 :112–147, 1998.

[62] J.L. Landa and R.N. Horne. A procedure to integrate well test data, reservoir performance history and 4d seismic data into a reservoir description. *SPE 38653, Proc. Annual Technological Conference and Exhibition, San Antonio, TX, 5-8 October*, 1997.

[63] H. Langouët. *Optimisation sans dérivées sous contraintes*. PhD thesis, Université de Nice-Sophia Antipolis, 2012.

[64] C.L. Lawson. *Solving Least Squares Problems*. SIAM Classics in Applied Mathematics, 1995.

[65] G. Li and A.C. Reynolds. Uncertainty quantification of reservoir performance predictions using a stochastic optimization algorithm. *Computational Geosciences*, 15(3) :451–462, 2011.

[66] N. Liu and D.S. Oliver. Automatic history matching of geologic facies. *SPE Journal*, 9(4) :429–436, 2004.

[67] G. Marsily, C. Lavedan, M. Boucher, and G. Fasanino. Interpretation of interference tests in a well field using geostatistical techniques to fit the permeability distribution in a reservoir model. *Geostatistics for Natural Resources Characterization*, 122(2) :831–849, 1984.

[68] B. Marteau, D. Ding, and L. Dumas. A generalized local gradual deformation method. *Proceedings of the 15th annual conference of the international association for mathematical geosciences*, 1 :699–702, 2013.

[69] B. Marteau, D.Y. Ding, and L.Dumas. A generalization of the local gradual deformation method using domain parameterization. *Computers & Geosciences*, 72(0) :233–243, 2014.

[70] C. Maschio and D.J. Schiozer. A new methodology for assisted history matching using independent objective functions. *Petroleum Science and Technology*, 26(9) :1047–1062, 2008.

[71] K.I.M. McKinnon. Convergence of the Nelder-Mead simplex method to a non stationary point. *SIAM Journal on Optimization*, 9 :148–158, 1998.

[72] A.D. Monfared, A. Helalizadeh, and H. Parvizi. Automatic history matching using the integration of response surface modeling with a genetic algorithm. *Petroleum science and technology*, 30(04) :360–374, 2012.

[73] P. Montleau, A. Cominelli, D.R.K. Neylon, I. Pallister, O. Tesaker, and I. Nygard. Production optimization under constraints using adjoint gradients. *Proceedings of the 10th European Conference on the mathematical oil recovery*, 2006.

[74] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7 :308–313, 1965.

[75] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer, 2006.

[76] A.A. Oberai1, N.H. Gokhale1, and G.R. Feijóo. Solution of inverse problems in elasticity imaging using the adjoint method. *Institute of Physics Publishing*, 19 :297–313, 2003.

[77] D.S. Oliver. Multiple realizations of the permeability field from well test data. *SPE Journal*, 1(2) :145–154, 1996.

[78] D.S. Oliver, A.C. Reynolds, and N. Liu. *Inverse theory for petroleum reservoir characterization and history matching*. Cambridge University Press, 2008.

[79] J.E. Onwunalu and L.J. Durlofsky. Application of a particle swarm optimization algorithm for determining optimum well location and type. *Computational Geosciences*, 14(1) :183–198, 2010.

[80] M. Parashar, H. Klie, U. Catalyurek, T. Kurc, W. Bangerth, V. Matossian, J. Saltz, and M.F. Wheeler. Application of grid-enabled technologies for solving optimization problems in data-driven reservoir studies. *Future Generation Computer Systems*, 21(1) :19 – 26, 2005.

[81] E. Peters, R.J. Arts, G.K. Brouwer, C.R. Geel, S. Cullick, R.J. Lorentzen, Y. Chen, K.N.B. Dunlop, F.C. Fossepoel, R. Xu, P. Sarma, A.H. Alhutali, and A.C. Reynolds. Results of the Brugge benchmark study for flooding optimization and history matching. *SPE Reservoir Simulation Symposium, the Woodlands, Texas, USA*, 2010.

[82] E. Peters, Y. Chen, O. Leeuwenburgh, and D.S. Oliver. Extended Brugge benchmark case for history matching and water flooding optimization. *Computers & Geosciences*, 50 :16–24, 2012.

[83] M. J .D. Powell. Least frobenius norm updating of quadratic models that satisfy interpolation conditions. *Mathematical Programming*, 100(1) :183–215, 2004.

[84] M. J .D. Powell. Developments of NEWUOA for minimization without derivatives. *IMA Journal of Numerical Analysis*, 28(4) :649–664, 2008.

[85] M.J.D. Powell. On trust region methods for unconstrained minimization without derivatives, 2002.

[86] M.J.D Powell. The NEWUOA software for unconstrained optimization without derivatives. *Large-scale Non Linear Optimization*, 83 :255–297, 2006.

[87] M.J.D Powell. On the convergence of trust region algorithms for unconstrained minimization without derivatives. *Computational Optimization and Applications*, 53(2) :527–555, 2012.

[88] B.S. RamaRao. Pilot point methodology for automated calibration of an ensemble of conditionally simulated transmitivity fields. *Theory and computational experiments*, 31 :475–493, 1995.

[89] M. Le Ravalec, B. Noetinger, and L.Y. Hu. The fft moving average generator : An efficient numerical method for generating and conditioning gaussian simulations. *Mathematical Geology*, 32(6) :701–723, 2000.

[90] L.C. Reis, L.Y. Hu, de G. Marsily, and R. Eschard. Production data integration using a gradual deformation approach : application to an oil field. *SPE Annual Technical Conference and Exhibition*, 2000.

[91] F. Roggero, D.Y. Ding, P. Berthet, O. Lerat, J. Cap, and P.E. Schreiber. Matching of production history and 4d seismic data - application to the girassol field, offshore angola. *SPE 109929, Proc. SPE ATC&E, Anaheim, California*, 2007.

[92] F. Roggero and L.-Y. Hu. Gradual deformation of continuous geostatistical models for history matching. *SPE Annual Technical Conference and Exhibition*, 1998.

[93] T. Romary. Integrating production data under uncertainty by parallel interacting markov chains on a reduced dimensional space. *Computational Geoscience*, 3 :103–122, 2009.

[94] I. Sahni and R.N. Horne. Multiresolution wavelet analysis for improved reservoir description. *SPE Reservoir Evaluation & Engineering*, 8(1) :53–69, 2005.

[95] I. Sahni and R.N. Horne. Generating multiple history-matched reservoir model realizations using wavelets. *SPE Reservoir Evaluation and Engineering*, 9(3), 2006.

[96] P. Sarma, W.H. Chen, L.J. Durlofsky, and K. Aziz. Production optimization with adjoint models under nonlinear control-state path inequality constraints. *SPE Journal*, 11(2) :326–339, 2008.

[97] P. Sarma and L. Durlofsky nad K. Aziz. Kernel principal component analysis for efficient, differentiable parameterization of multipoint geostatistics. *Mathematical Geosciences*, 40(1) :3, 2008.

[98] K. Scheinberg and P.L. Toint. Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization. *SIAM Journal on Optimization*, 20(6) :3512–3532, 2010.

[99] D.F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation*, 24(111) :647–656, 1970.

[100] D. Sinoquet and F. Delbos. Adapted nonlinear optimization method for production data and 4d seismic inversionnormal access. *11th European Conference on the Mathematics of Oil Recovery*, 2008.

[101] J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37, 1992.

[102] J.C. Spall. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Automatic Control*, 34(3) :817–823, 1998.

[103] P. Spellucci. An SQP method for general nonlinear programs using only equality constrained subproblems. *Mathematical Programming*, 82(3) :413–448, 1998.

[104] R.W. Stallman. Numerical analysis of regional water levels to define aquifer hydrology. *Transacions, American Geophysical Union*, 37(4) :451–460, 1956.

[105] Z. Tavassoli, J.N. Carter, and P.R. King. Errors in history matching. *SPE Journal*, 9(3) :352–361, 2004.

[106] V. Torczon. On the convergence of search algorithms. *SIAM Journal on Optimization*, 7(1) :1–25, 1997.

[107] A. Tröltzsch. *An active-set trust-region method for bound-constrained nonlinear optimization without derivative*. PhD thesis, Université de Toulouse, 2011.

[108] A. Vardi. A trust region algorithm for equality constrained minimization : convergence properties and implementation. *SIAM Journal of Numerical Analysis*, 22 :575–591, 1985.

[109] C. Wang, G. Li, and A.C. Reynolds. Production optimization in closed-loop reservoir management. *SPE journal*, 14(03) :506–523, 2009.

[110] P. Wang. *Development and applications of production optimization techniques for petroleum fields*. PhD thesis, Standford University, 2003.

[111] S. Yadav, R. Heim, S. Bryant, and R. Sinha. Optimal region delineation in a reservoir for efficient history matching. *SPE Annual Technical Conference and Exhibition*, 2007.

[112] H. Zhang and A.R. Conn. On the local convergence of a derivative-free algorithm for least-squares minimization. *Computational Optimization and Applications*, 51(2) :481–507, 2012.

[113] H. Zhang, A.R. Conn, and K. Scheinberg. A derivative free-algorithm for least-squares minimization. *SIAM Journal on Optimization*, 20(6) :3555–3576, 2010.

[114] H. Zhao, C. Chen, S. Do, G. Li, and A.C. Maximization of a dynamic quadratic interpolationmodel for production optimization. *SPE International*, 2011.

[115] H. Zhao, C. Chen, S. Do, D. Oliveira, G. Li, and A. Reynolds. Maximization of a dynamic quadratic interpolation model for production optimization. *SPE Journal*, 18(6) :1012–1025, 2013.

[116] H. Zhao, G. Li, A.C. Reynolds, and J. Yao. Large-scale history matching with quadratic interpolation models. *Computational geosciences*, 17(1) :117–138, 2013.

Table des figures

1 Boucle illustrant le calibrage des paramètres d'un modèle par calage d'historique. 6
2 Calage de la pression au fond d'un puits producteur du réservoir Brugge avant et après calage d'historique. 7
3 Illustration de la séparabilité partielle de la fonction objectif associée à champ synthétique. 9

1.1 Combinaison de deux bruits blancs Gaussiens par déformation graduelle. 14
1.2 variations d'un modèle avec un paramètre de déformation graduelle ; de (a) à (j), ρ^1 varie d'un pas constant ϵ. 15
1.3 Combinaison locale de deux bruits blancs Gaussiens par déformation graduelle. . 16
1.4 Modèle de réservoir dans lequel le bruit blanc Z_1 est appliqué dans R_1 et Z_0 hors de R_1. 16
1.5 Illustration de la méthode de déformation de domaines lorsque plusieurs domaines sont définis. 17
1.6 Comparaison entre la méthode de déformation de domaines et la méthode de déformation graduelle locale suivant la taille des régions [37]. 18
1.7 Séparation d'un réservoir en 11 zones supposées indépendantes. 23
1.8 Illustration de la méthode de patchwork classique. 25
1.9 Illustration de la méthode de patchwork améliorée. 26
1.10 Carte de perméabilité du champ synthétique comprenant 13 puits producteurs et 12 puits injecteurs [37]. 27
1.11 Initialisation par méthode de patchwork améliorée. 29
1.12 Valeur de la fonction objectif pour la meilleure réalisation (en bleu) et pour la réalisation construite via patchwork avec des rayons de $r = 300m$ (en rouge) pour les 5 cas considérés. 30
1.13 Valeur de la fonction objectif en fonction du rayon des zones pour le cas de figure 5. 30
1.14 Comparaison des méthodes de déformation graduelle locale et de déformation graduelle généralisée sur les 5 cas présentés. 32
1.15 Comparaison des méthodes de déformation de domaines et de déformation graduelle généralisée sur les 5 cas présentés. 33

2.1 Exemple d'une fonction et d'un ensemble d'interpolation qui mène l'algorithme à construire des modèles dégénérés (issu de [98]). 46
2.2 Un ensemble d'interpolation présentant une mauvaise Λ-unisolvance (a) et un autre présentant un bonne Λ-unisolvance (b) (voir [27]). 49
2.3 Les croix sont les points à évaluer pour construire un modèle quadratique avec $2p+1$ points sans exploiter la séparabilité partielle tandis que les carrés sont ceux utilisés pour construire les modèles de f_1 et f_2. 60
2.4 Graphe de dépendance de la fonction F définie en (2.22). 61
2.5 Graphe de dépendance coloré de la fonction F (2.22). 61

2.6 Nombre d'évaluations de la fonction objectif pour arriver à convergence en fonction de la dimension du problème pour chacune des fonctions tests avec l'algorithme DFO-PSOF. 75
2.7 Champ synthétique comprenant 13 puits producteurs et 12 puits injecteurs [37]. . 76
2.8 Influence du paramètre associé au puits PRO_7. 77
2.9 Comparaison de calages d'historique par les méthodes SQA et DFO-PSOF. . . . 78

3.1 Perméabilité de la première couche du réservoir PUNQ avec l'emplacement des puits. 80
3.2 Carte de perméabilité de la première couche du modèle de référence du cas Brugge. 81
3.3 Emplacement des zones créées par la méthode de patchwork améliorée. 83
3.4 Valeur de la fonction objectif après patchwork en fonction de la taille des zones pour le cas PUNQ. 84
3.5 Découpage du modèle de réservoir pour un patchwork classique. 85
3.6 Optimisation de la fonction objectif par déformation graduelle classique et par déformation graduelle généralisée sur le cas PUNQ. 85
3.7 Optimisation de la fonction objectif pour les 3 variantes de la méthode de déformation graduelle généralisée. 86
3.8 Répartition des zones créées par la méthode de patchwork améliorée sur le réservoir Brugge. 89
3.9 Évolution de la valeur de la fonction objectif en fonction de la taille des zones sur le réservoir Brugge. 89
3.10 Découpage classique du réservoir Brugge en zones délimitées par les mailles du modèle. 90
3.11 Optimisation de la fonction objectif en utilisant la méthode de déformation graduelle classique avec la réalisation de base comme point initial (en bleu), avec la réalisation obtenue par patchwork (en vert) et la méthode de déformation graduelle généralisée (en rouge) sur le cas Brugge. 91
3.12 Calage du Water Cut (WC) sur le puits P-2 après optimisation par déformation graduelle classique et généralisée, cas Brugge. 92
3.13 Optimisation de la fonction objectif par les trois variantes de la méthode de déformation graduelle généralisée (valeur initiale des rayons à 525m), cas Brugge. 92
3.14 Optimisation de la fonction objectif par les 3 variantes de la méthode de déformation graduelle généralisée (valeur initiale des rayons à $225m$), cas Brugge. . . . 93
3.15 Calage du WC pour les puits P-10, P-13 et P-15, cas Brugge. 95
3.16 Comparaison des trois variantes de la méthode de déformation graduelle généralisée sur une seconde optimisation, cas Brugge. 96
3.17 Calages des données de production pour le puits producteur P-17 (BHFP à gauche et WC à droite), cas Brugge. Le modèle initial est le résultat de la première optimisation. 97
3.18 Calages d'historiques réalisés avec les méthodes SQA et DFO-PSOF, cas PUNQ. 99
3.19 Calages d'historiques réalisés avec les méthodes d'optimisation SQA et DFO-PSOF sur les trois variantes de la méthode de déformation graduelle généralisée : (a) aucun paramètre de rayon ; (b) un paramètre de rayon ; (c) 13 paramètres de rayon, cas PUNQ 100
3.20 Calage de l'historique de production du puits PRO-2, cas PUNQ. 101
3.21 Cartes des perméabilités du réservoir, cas PUNQ. 102
3.22 Cartes des perméabilités du réservoir, cas PUNQ. 103
3.23 Cartes des saturations en huile du réservoir, cas PUNQ. 103

3.24 Cartes des saturations en huile du réservoir, cas PUNQ. 104
3.25 Comparaison entre SQA et DFO-PSOF dans le cas où aucun paramètre de rayon n'a été introduit, cas Brugge. 106
3.26 Comparaison entre SQA et DFO-PSOF dans le cas où sept paramètres de rayon indépendants ont été introduits, cas Brugge. 106
3.27 Comparaison entre SQA et DFO-PSOF dans le cas où aucun paramètre de rayon n'a été introduit, cas Brugge. 107
3.28 Comparaison entre SQA et DFO-PSOF dans le cas où un unique paramètre de rayon a été introduit, cas Brugge. 108
3.29 Comparaison des algorithmes SQA et DFO-PSOF sur le deuxième calage du cas Brugge. 110
3.30 Calages des données de production pour le puits producteur P-2, cas Brugge. . . 110
3.31 Calages des données de production pour le puits producteur P-10, cas Brugge. . . 111
3.32 Calages des données de production pour le puits producteur P-6, cas Brugge. . . 111
3.33 Cartes de perméabilité pour les réalisations de référence (à gauche), de base (au milieu) et optimale (à droite),cas Brugge. 112
3.34 Cartes de saturaton en huile de la couche superficielle du réservoir pour les réalisations de référence (à gauche), de base (au milieu) et optimale (à droite), cas Brugge. 113

Table des algorithmes

1 Algorithme général de recherche directe 41
2 Algorithme Nelder-Mead 43
3 Un algorithme de type région de confiance 44
4 Un algorithme simpliste de type région de confiance avec modèle d'interpolation . 45
5 Un algorithme d'amélioration de la Λ-unisolvance 54
6 Algorithme NEWUOA ou SQA (variante IFPEN) 55
7 Un algorithme sans dérivées utilisant une propriété d'auto-correction des ensembles d'interpolation 57
8 Algorithme d'optimisation sans dérivées adapté aux fonctions partiellement séparables (DFO-PSOF). 64
9 Gestion de l'ensemble d'interpolation des sous-modèles à l'itération k de l'algorithme 8. 65

Résumé

Dans cette thèse, on s'intéresse à un problème inverse classique en ingénierie pétrolière, à savoir le calage d'historique. Plus précisément, une nouvelle méthode de paramétrisation géostatistique ainsi qu'un nouvel algorithme d'optimisation sans dérivées adaptés aux particularités du problème sont présentés ici.

La nouvelle méthode de paramétrisation repose sur les principes des méthodes de déformation graduelle et de déformation de domaines. Comme la déformation graduelle locale, elle consiste à combiner à l'intérieur de zones préalablement définies deux réalisations ou plus de modèle avec la possibilité supplémentaire de modifier dynamiquement la forme des zones choisies. La flexibilité apportée par cette méthode dans le choix des zones a ainsi permis de garantir l'obtention d'un bon point initial pour l'optimisation.

Concernant l'optimisation, l'hypothèse que les paramètres locaux dans le modèle de réservoir n'influent que faiblement sur les données de puits distants conduit à considérer que la fonction à optimiser est à variables partiellement séparables. La nouvelle méthode d'optimisation développée, nommée DFO-PSOF, de type région de confiance avec modèle quadratique d'interpolation, exploite alors au maximum cette propriété de séparabilité partielle. Les résultats numériques obtenus sur plusieurs cas de réservoir valident à la fois l'hypothèse effectuée ainsi que la qualité de l'algorithme pour le problème de calage d'historique. En complément de cette validation numérique, un résultat théorique de convergence vers un point critique est prouvé pour la méthode d'optimisation construite.

Mots-clés : paramétrisation, géostatistiques, optimisation sans dérivées, séparabilité partielle.

Abstract

We worked in this thesis on a classical inverse problem in the petroleum industry, history matching. We proposed a new geostatistical parameterization technique as well as a new derivative free optimization algorithm adapted to the problem specificities.

The parameterization method is based on two approaches found in the literature, the local gradual deformation method and the domain deformation method. Similarly to the local gradual deformation method, our method combines two or more model realizations inside previously defined zones. Moreover, our method adds the possibility to dynamically update the shape of the zones during the optimization process. This property substantially improves its robustness with regard to the initial choice of the zones. Thus, the greater flexibility brought by our method allowed us to develop an initialization methodology which garantees a good initial point for the optimization.

To reduce the number of evaluations needed to minimize the objective function, we made the assumption that a local parameter does not influence the production data of a distant well. With this hypothesis, the objective function is then considered partially separable. The optimization algorithm we developped, called DFO-PSOF, is a trust region algorithm based on quadratic interpolation models which exploits this partial separability property. Numerical results obtained on some reservoir test cases validate both the hypothesis and the quality of our algorithm for the history matching problem. Moreover, a theoretical convergence result towards a first order critical point, is proved for this new optimization method.

Key-words : parameterization, geostatistics, derivative free optimization, partial separability.