Restoration and separation of piecewise polynomial signals. Application to Atomic Force Microscopy

Junbo Duan

Nancy-Université

École Doctorale IAEM Lorraine

CRAN

THÈSE

présentée pour l'obtention du

Doctorat de l'Université Henri Poincaré Nancy 1
(Spécialité Automatique, Traitement du Signal et Génie Informatique)

par

Junbo DUAN

Restauration et séparation de signaux polynômiaux par morceaux
Application à la microscopie de force atomique

Soutenue publiquement le 15^{er} novembre 2010

Composition du jury

Président :	R. LENGELLE	Professeur à l'Université Technologie de Troyes (UTT, Troyes)
Rapporteurs :	J. MARS	Professeur à l'INP Grenoble, (GIPSA-LAB, Grenoble)
	C. HEINRICH	Professeur à l'Université Louis Pasteur (LSIIT, Strasbourg)
Examinateurs :	D. BRIE	Professeur à l'Université Henri Poincaré, Nancy 1 (CRAN, Nancy)
	C. SOUSSEN	MCF à l'Université Henri Poincaré, Nancy 1 (CRAN, Nancy)
	J. IDIER	Directeur de recherche au CNRS (IRCCyN, Nantes)
Invité :	G. FRANCIUS	Chargé de Recherche au CNRS (LCPME, Nancy)

À mon père et ma mère.

Remerciements

Le travail présenté dans cette thèse a été réalisé au sein du Groupe Théatique Identification, Restauration, Images, Signaux (IRIS) du Centre de Recherche en Automatique de Nancy (CRAN). Tout d'abord je tiens à remercier Messieurs Alain Richard et Hugues Garnier, directeur du CRAN et responsable du IRIS respectivement, pour m'avoir accueilli dans le laboratoire. Je remercie Monsieur Francis Lepage, vice-président de l'UHP en charge des relations internationales, pour avoir facilité mon accueil.

J'exprime ma profonde gratitude à Messieurs Jérôme Mars et Christian Heinrich, pour avoir accepté de rapporter ma thèse écrite en français, anglais et "franglish". Je remercier également Monsieur Régis Lengellé pour avoir examiné mon travail et présidé mon jury de soutenance.

Monsieur David Brie a dirigé mon travail pendant ces trois années, un directeur de thèse avec vue perçante, pensée dynamique et coeur compéhensif, je tiens à vous dire "Merci mon chef, chef un jour, chef toujours." Monsieur Charles Soussen, co-directeur de ma thèse, collègue scolaire et frère quotidien, m'a donné des aides et avis incalculable au long de mon séjour en france, *e.g.,* aide dans différentes démarches administratives, rédaction de documents mais aussi pour les loisirs partagés ensemble (invitation au "théatre ça respire encore",... *etc.*). Merci pour tout, Chuck.

Monsieur Jérôme Idier, mon autre co-encadrant, un chercheur à temps plein avec rigueur et passion à la fois, je tiens à le remercie aussi pour m'avoir invité et accueilli à Nantes lors de mon séjour doctoral, et pour les discussions constructives que nous avons eu.

Je remercie également Messieurs Grégory Francius, Pavel Polyakov et Fabien Gaboriaud, mes collègues du Laboratoire de Chimie-Physique et Microbiologie pour l'Environnement (LCPME), pour l'intérêt qu'ils ont porté à mon travail, pour tous les échanges que nous avons eu et pour les nombreuses acquisitions de données expérimentales en lien avec mon travail.

Je souhaite remercier Messieurs Sebastian Miron, El-Hadi Djermoune, Thierry Bastogne et Madame Marion Gilson, membres du groupe IRIS, pour leurs aides aimables. Remerciement spécial à notre secrétaire, Madame Sabine Huraux, pour son efficacité dans les tâches administratives.

Merci également à mes compagnons de bureau : Simona Dobre, Fabrice Caland, Abdelhamid Bennis, Jean-Baptiste Tylcz, pour leur accompagnement tout au long de mon séjour au CRAN. Remerciement spécial à Xijing Guo, pour m'avoir aidé à trouver une thèse en France.

Merci à mes copains Abdouramane Moussa Ali, Tushaar Jain, Adriana Aguilera, pour le temps passé ensemble, espécialement durant les déjeuner au Resto U. Merci aux doctorants et docteurs du CRAN : Dragos Dobre, Hossein Hashemi Nejad, Sinuhe Martinez, Souleymen Sahnoun, Vincent Laurain, Kamel Menighed, Ahmed Khelassi, Boumedyen Boussaid, Simon Herrot et Julien Schorsh, pour leurs conseils professionnelles et privés.

Merci enfin à tous mes amis chinois : Xianqing Mao, Xiaojun Li, Lifan Zou, Zhen Wang, Xiaobai Zhou, Tingting Ding, Feng Zou, Min Chen, Feng Zhen, Na Du, Yuanyuan Guo, Pei Li, Yan Lu, Jing Yang, Zhijie Wang, Xiang Yan, Haitao Liu, Ping Lan, Li Yi, Lin Zhang, Huahua Chen, et mes amis étrangers : Caloi Tang, Neeraj Kumar Singh, Ashish Malik, Andrew Ngadin, avec qui je garde de très bon souvenir de Nancy.

Table des matières

Chapitre 1

Introduction générale

Cette thèse est rédigée "par articles". Le présent chapitre constitue une introduction à la microscopie de force atomique, application qui motive les développements réalisés, et une présentation synthétique des contributions méthodologiques et appliquées de cette thèse. Ce chapitre est le chapitre principal autour duquel s'articulent les autres chapitres qui décrivent plus en détail chacun des algorithmes développés. Les chapitres 2 et 3 sont dédiés aux algorithmes d'approximation parcimonieuse. Le chapitre 4 concerne le problème de la séparation de sources parcimonieuses à partir de mélanges retardés. Enfin, le chapitre 5 présente un outil dédié au traitement des courbes de force et des images force-volume acquises en microscopie AFM, basé sur l'utilisation des modèles physiques existants.

Dans ce chapitre introductif, nous commençons par présenter la microscopie de force atomique (AFM), qui est une technologie récente permettant de mesurer des forces d'interaction entre nano-objets. Nous décrivons le type de signaux et d'images que l'on peut acquérir à l'aide d'un microscope AFM, et le type d'information physique que l'on cherche à extraire de ces données.

Dans la deuxième partie de ce chapitre, nous décrivons les problématiques du sujet de thèse en terme de problèmes inverses de traitement du signal. L'estimation de paramètres physiques à partir de courbes de force et d'images force-volume mesurées en microscopie de force atomique motive les développements algorithmiques effectués. Les traitement envisagés sont basés sur la notion de *parcimonie*, au sens où les courbes de force contiennent quelques points caractéristiques (sauts, changements de pente, changements de courbure) qu'il s'agit d'estimer très précisément. Les problèmes inverses étudiés se résument en la *restauration de signaux parcimonieux* (lissage d'une courbe de force) et la séparation de ces signaux à partir d'un grand nombre de mélanges retardés.

Dans la troisième partie du chapitre, nous présentons très succinctement les contributions algorithmiques et nous les illustrons par quelques résultats types obtenus sur des données réelles.

La dernière partie du chapitre résume l'organisation des autres chapitres de la thèse.

1.1 Contexte applicatif et objectifs

1.1.1 Contexte général

Cette thèse a pour cadre un projet collaboratif entre le CRAN, le LCPME et l'IRCCyN. Elle a pour objectif le développement d'outils originaux de traitement des images de force atomique qui intègrent le savoir-faire des laboratoires, à savoir la spectroscopie moléculaire, et plus particulièrement la microscopie de force atomique pour le LCPME, et le traitement du signal et de l'image pour le CRAN et l'IRCCyN.

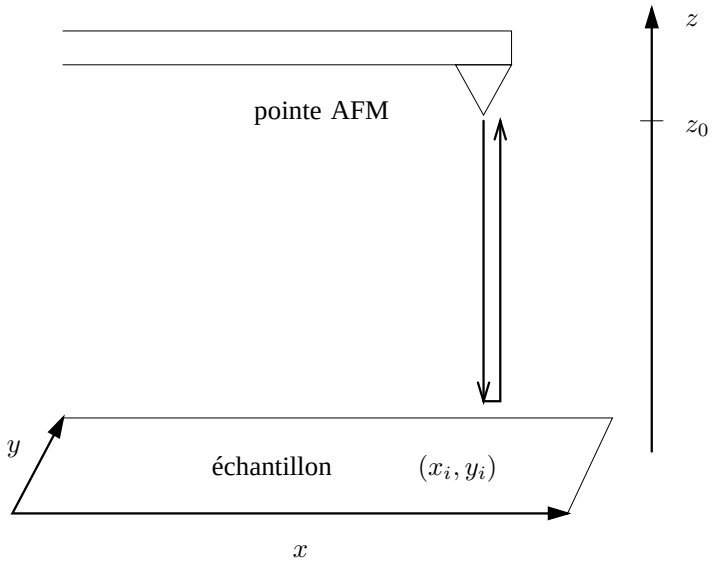


FIGURE 1.1 – Spectroscopie de force. En un point de l'échantillon, la pointe AFM s'approche jusqu'au contact déformable puis non déformable de l'échantillon (phase d'approche), puis la pointe se retire et perd contact avec l'échantillon (phase de retrait). Dans chaque phase, un signal $f(z)$ est mesuré. Ces signaux sont appelés courbes d'approche et de retrait.

L'étude des forces interatomiques et intermoléculaires a une longue histoire ; elle est étroitement liée à la compréhension des processus se déroulant aux interfaces solide/solution aqueuse. Le développement des microscopies à champ proche, et en particulier de la microscopie à force atomique (AFM), permet à l'heure actuelle de déterminer *in situ* des propriétés physico-chimiques locales (électriques, magnétiques, vibrationnelles, forces) [1]. Ces techniques fournissent des mesures spectroscopiques à l'échelle nanométrique indépendamment de la nature des échantillons (biologique, organique, minérale), et des images tridimensionnelles, dites *images force-volume* en microscopie AFM [2].

1.1.2 Microscopie AFM

Le principe de fonctionnement d'un microscope AFM est basé sur la détection des forces interatomiques (capillaires, électrostatiques, Van der Waals, frictions) s'exerçant entre une pointe associée à un levier de constante de raideur fixe, et la surface d'un échantillon [2]. On distingue généralement deux modes d'acquisition des données.

Images isoforce et isodistance

Ces données résultent du balayage par la pointe de toute la surface de l'échantillon, selon deux modes opératoires :

- le mode contact (ou statique). Dans ce cas, un système d'asservissement permet de maintenir la force exercée sur le levier supportant la pointe au cours du balayage. Cette force est proportionnelle à la déflexion du levier qui est la grandeur mesurée. Le mode contact permet d'obtenir directement la topographie, c'est-à-dire le relief des surfaces ;
- le mode sans contact (ou vibrant). Typiquement, une variation des forces d'interaction se traduit par une variation de la fréquence de résonance du levier qui est la grandeur mesurée. Par inversion de la relation liant la force à la variation de fréquence, on obtient une image de la force. Dans ce cas, le système d'asservissement permet de maintenir soit la distance entre la

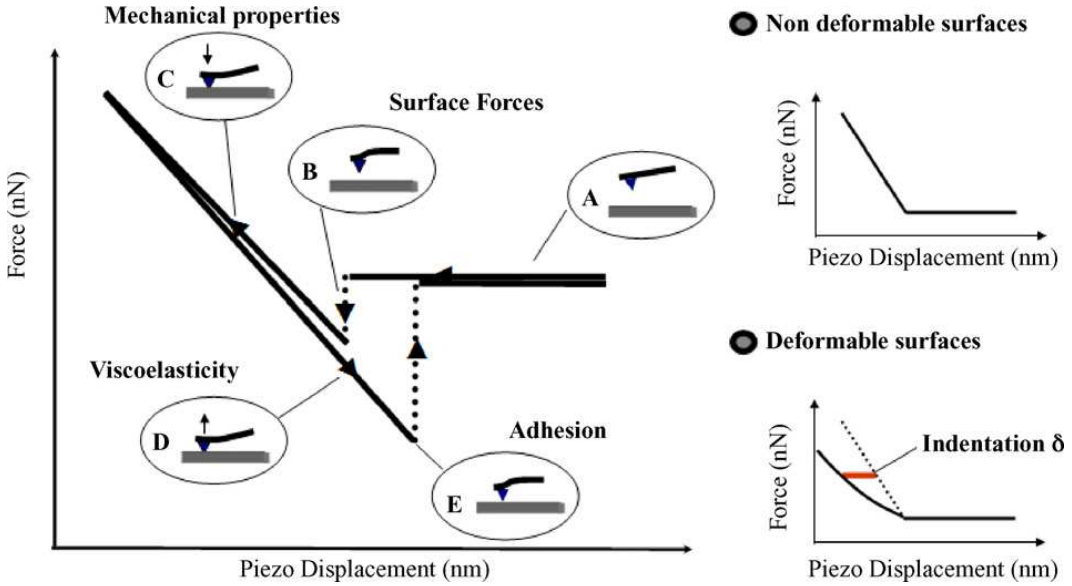
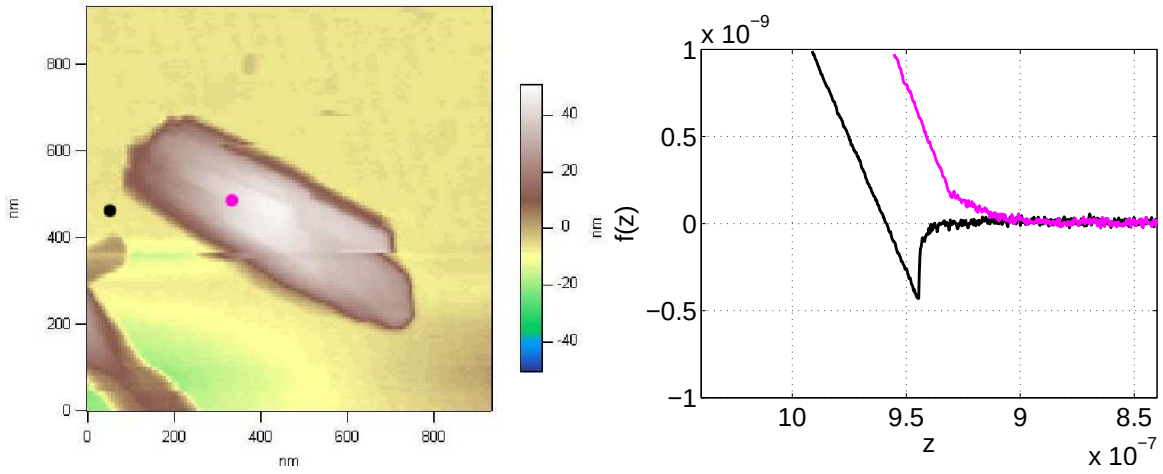


FIGURE 1.2 – Forme générale d'une courbe de force. Courbes d'approche (en trait continu) et de retrait (en traits pointillés) de la pointe. Adapté de [3].



(a) (b)

FIGURE 1.3 – Application du mode force-volume pour une particule nanométrique de goethite (α-FeOOH) déposée sur une lame de verre en interaction avec une pointe AFM recouverte d'un oxyde d'aluminium chargé positivement (pH = 4, $NaNO_3$ = 1 mM). (a) Image AFM obtenue en mode contact en milieu liquide. (b) Acquisition de courbes de force mesurées soit en surface de la particule de goethite (interactions répulsives) soit sur le support (interactions attractives).

pointe et l'échantillon soit la force agissant sur le levier. Ce mode permet donc d'obtenir des images isodistance ou isoforce.

Spectroscopie de force

Contrairement au mode d'acquisition précédent, la spectroscopie de force est une analyse *ponctuelle* de l'échantillon, obtenue en enregistrant la déflexion du levier en fonction de la distance z entre la pointe et la surface de l'échantillon (voir la Fig. 1.1). **Une courbe de force $f(z)$ montre donc l'évolution de la force en fonction de la distance z en un point de l'échantillon**.

La forme générale d'une courbe de force est présentée sur la Fig. 1.2. L'intensité de la force est calculée à partir de la mesure relative de la déflexion du levier, en fonction du déplacement relatif de la pointe z, où les plus grandes valeurs de la distance z correspondent aux positions de la pointe les plus éloignées de l'échantillon. Dans une courbe de force, on distingue deux courbes, correspondant à l'approche et au retrait de la pointe (en trait continu et en traits pointillés respectivement). Dans ce qui suit, nous décrivons des zones particulières sur ces courbes.

Courbe d'approche :
- *Zone A.* Aucune interaction n'est observée lorsque la pointe est placée à une distance suffisamment grande de l'échantillon. Cette zone permet de normaliser les courbes en définissant cette valeur comme la force nulle. Rappelons qu'une courbe de force est obtenue de manière relative en abscisse (déplacement de la pointe) et en ordonnée (force) ;
- *Zone B.* Elle correspond aux forces de surface (électrostatiques, Van der Waals). Ces interactions sont négatives (attraction entre la pointe et la surface) ou positives (répulsion). Dans cette zone, la pointe n'est pas en contact avec l'échantillon ;
- Entre les zones B et C se situe le point de contact entre la pointe et l'échantillon ;
- *Zone C.* Interactions mécaniques du levier et/ou de l'échantillon. Pour un échantillon non déformable, le comportement observé est du essentiellement à la déformation linéaire du levier. En revanche, pour un échantillon déformable, des processus de compression et/ou indentation de l'échantillon conduisent à des comportements linéaires ou non linéaires.

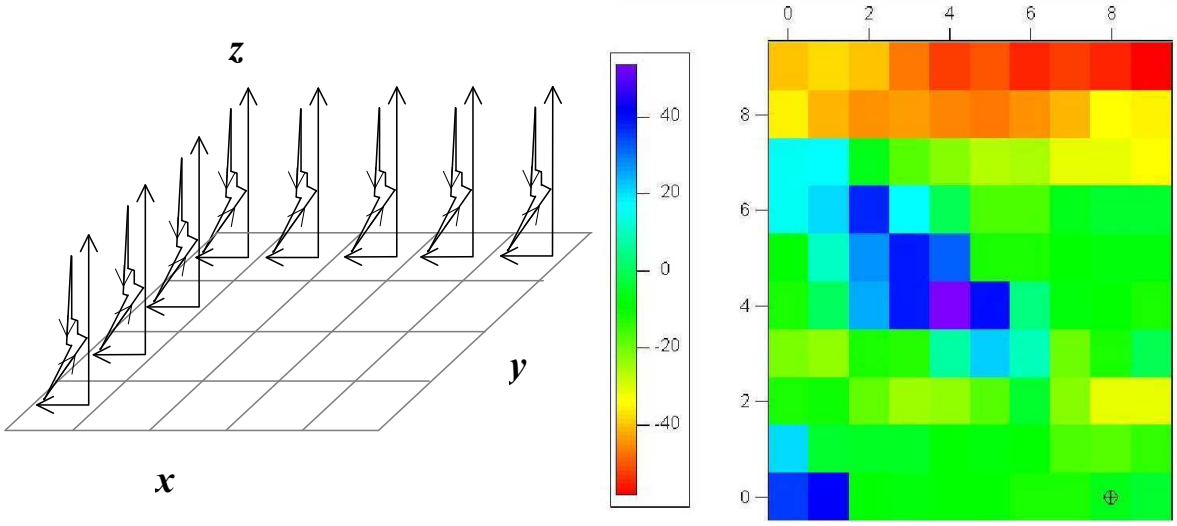
Courbe de retrait :
- *Zone D.* La présence d'une hystérèse entre les courbes d'approche et de retrait traduit les propriétés viscoélastiques de l'échantillon. Pour les surfaces non déformables, cette hystérèse est nulle ;
- *Zone E.* Au cours de ce retrait, les courbes peuvent présenter une force de rappel importante en fonction de l'aire et du temps de contact mais aussi et surtout de l'énergie de surface échantillon-pointe. Sur des surfaces de micro-organismes, ce domaine présente de nombreuses ruptures qu'il convient d'analyser.

La Fig. 1.3 (b) présente des courbes de force mesurées pour une particule nanométrique de goethite (α-FeOOH) déposée sur une lame de verre. Les deux courbes de force présentées correspondent à deux points de la surface de l'échantillon, l'un à l'intérieur et l'autre à l'extérieur de la particule. On constate que la topographie est différente en ces deux points, car le contact pointe-échantillon (zone B-C des courbes) n'est pas obtenu pour les mêmes valeurs de z. D'autre part, les forces surfaciques sont alternativement répulsives et attractives pour ces deux courbes.

Imagerie force-volume

En reproduisant l'analyse ponctuelle précédente et en balayant la surface de l'échantillon, on obtient une *image force-volume* $f(x, y, z)$. Cette image résulte de la collection des courbes de force $f(z)$ sur une grille (x, y) représentant la surface de l'échantillon (voir Fig. 1.4 (a)).

Il n'est pas aisé de visualiser cette image tridimensionnelle. Un traitement partiel consiste à estimer le point de contact entre la pointe et l'échantillon pour une courbe de force donnée. La



(a) (b)

FIGURE 1.4 – Imagerie force-volume. (a) Acquisition d'une image force-volume sur une grille (x, y) représentant la surface de l'échantillon. (b) Application à l'imagerie d'une nano-particule de goethite. Construction d'une image 2D représentant le point de contact pointe-échantillon en chaque point de la surface, à partir d'une image force-volume acquise sur 10×10 pixels de la surface de l'échantillon.

répétition de ce traitement en chaque pixel (x_i, y_i) permet de construire une image 2D représentant la topographie de l'échantillon. Cette image est présentée sur la Fig. 1.4 (b) dans le cas de la nano-particule de goethite.

1.1.3 Objectifs du projet

L'objectif du projet est de développer des outils originaux de traitement du signal adaptés au traitement des images force-volume. Ces traitements consistent en :

1. la reconstruction de la topographie de nano-objets. C'est un problème difficile qui n'a pas reçu, à notre sens, de solution satisfaisante. Sa résolution permettra également d'envisager des avancées dans l'interprétation et l'exploitation des données fournies par d'autres techniques de microscopie à champ proche (optique notamment).

2. la recherche d'interaction élémentaires récurrentes dans les courbes de force qui constituent une image force-volume. Il s'agit de les identifier et d'estimer leur répartition relative dans le mélange par des techniques de séparation de sources.

Les méthodes reposeront sur l'utilisation des modèles physiques disponibles pour décrire les forces inter-atomiques. Le champ d'applications, potentiellement très large, se restreint dans ce projet à l'étude de systèmes microniques ou submicroniques rencontrés dans l'environnement, des particules minérales colloïdales à propriétés chimiques de "surface" hétérogènes évolutives jusqu'aux micro-organismes de type bactéries ou virus dont la physico-chimie de surface est loin d'être cernée [3], [4].

La suite de ce document présente de façon plus détaillée le sujet de thèse et les principales contributions

1.2 Traitement de données et besoins algorithmiques

1.2.1 Traitement de courbes de force et d'images force-volume

En microscopie AFM, l'interprétation des images force-volume reste essentiellement descriptive. Le travail de thèse a pour but de concevoir des méthodes automatiques de traitement, pour caractériser très finement la surface de l'échantillon analysé à partir d'une image force-volume.

Comme nous l'avons déjà précisé, la spectroscopie de force est une analyse ponctuelle de la surface d'un échantillon. Elle fournit une collection de mesures $f(z)$ de la force s'exerçant entre la pointe et la surface de l'échantillon, en fonction de la distance pointe-échantillon z [1]. Une image force-volume $f(x, y, z)$ correspond au balayage de l'échantillon. Elle résulte de la collection des courbes de force mesurées sur une grille (x, y) représentant la surface de l'échantillon.

Nous envisageons deux types de traitements sur une image force-volume, qui correspondent en fait à deux modèles différents.

Lissage d'une courbe de force et extraction de paramètres physiques

Dans cette approche, chaque courbe de force est traitée séparément et indépendamment des autres courbe de l'image force-volume. Une courbe de force est décrite par un ensemble de **modèles paramétriques par morceaux**. Typiquement, pour une courbe d'approche, on observe plusieurs types d'interaction successivement : une interaction de type électrostatique avant contact puis des interactions mécaniques après contact. À chaque type d'interaction est associé un modèle paramétrique connu (voir la Fig. 1.5). Pour pouvoir extraire des informations physiques d'une courbe de force (topographie de l'échantillon, énergie d'adhésion, *etc.*), il est essentiel de détecter les régions d'intérêt (les morceaux) où les modèles paramétriques s'appliquent. C'est la principale difficulté du traitement. Les logiciels disponibles pour analyser les courbes de force reposent sur une détection des régions d'intérêt suivant des règles empiriques.

Dans cette thèse, nous étudions des stratégies pseudo-automatiques pour la détection des régions d'intérêt. Le terme "pseudo-automatique" indique que l'on cherche à développer des algorithmes aussi automatiques que possible, mais il est parfois nécessaire d'introduire quelques paramètres de réglage comme un seuil sur le l'erreur d'approximation lors du lissage d'une courbe de force en utilisant un modèle paramétrique. La Fig. 1.5 illustre le fait que pour les courbes d'approche comme pour les courbes de retrait, les points qui se trouvent à l'extrémité de chaque région d'intérêt sont des **points de discontinuité**, où la courbe de force possède un saut et/ou un changement de pente et/ou un changement de courbure. Dans le chapitre 4, nous proposons un algorithme en deux étapes, qui repose sur la détection des régions d'intérêt où les modèles physiques s'appliquent, et l'estimation des paramètres de chaque modèle par moindres carrés sur chacune des régions d'intérêt. L'application de l'algorithme à toutes les courbes de force d'une image force-volume permet la reconstruction d'un ensemble d'images 2D représentant chacune l'un des paramètres physiques.

Séparation de sources retardées

Nous proposons finalement une autre approche qui modélise une courbe de force comme un mélange de signaux parcimonieux retardés (appelés *signaux sources*). La courbe de force mesurée

1. Les signaux et images acquis en microscopie AFM sont discrets, mais nous préférons, par simplicité, utiliser des notations continues pour les grandeurs x, y, z, sauf quand la formulation discrète est vraiment nécessaire.

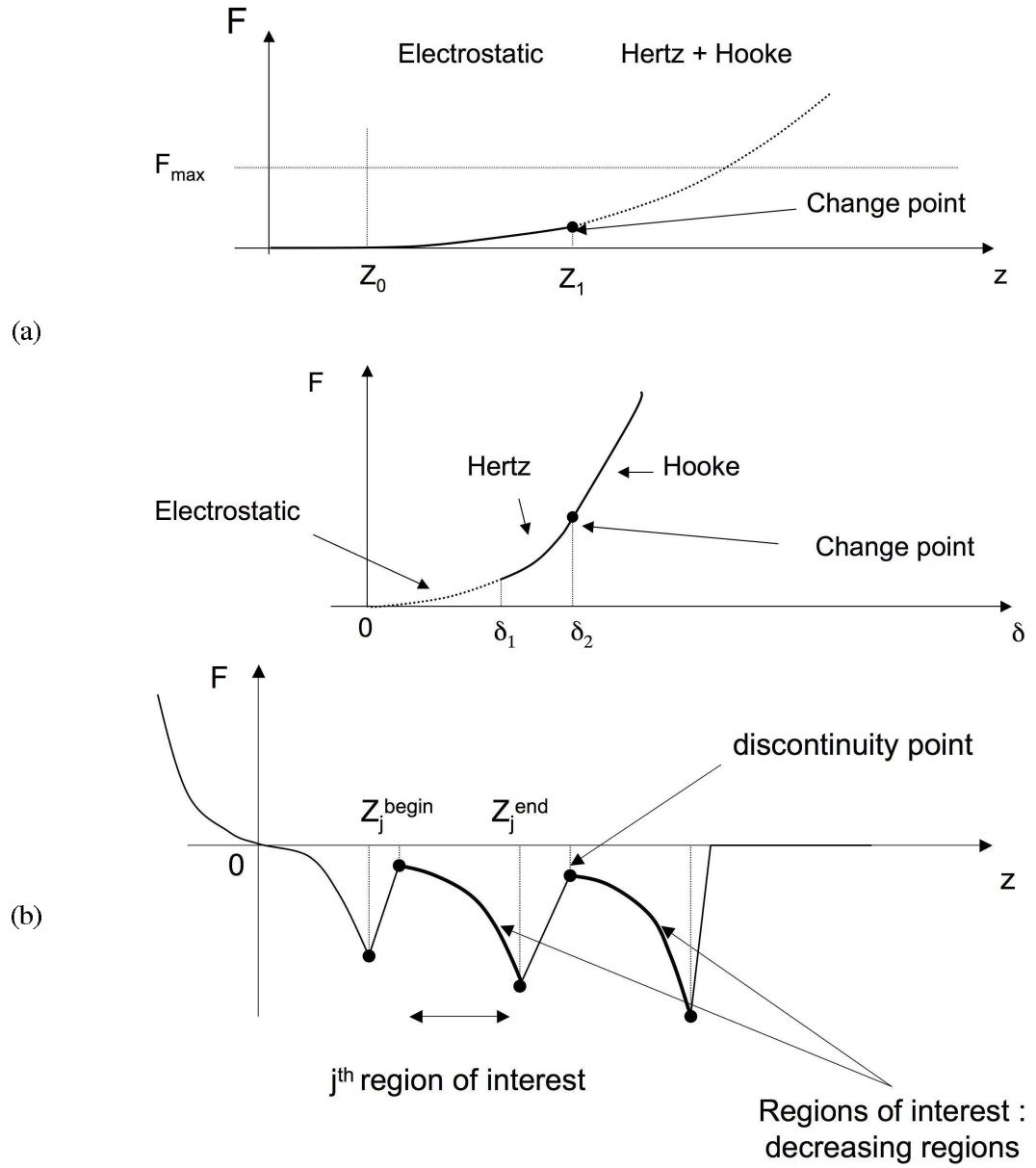


FIGURE 1.5 – Modèles par morceaux pour les courbes d'approche et de retrait. (a) Sur une courbe d'approche, on peut observer trois régions d'intérêt, définies dans deux domaines différents (courbes $z \mapsto f(z)$ et $\delta \mapsto f(\delta)$, où $\delta = z - Z_0 - (F - F_0)/k_c$ et k_c est la constante de raideur du levier sur lequel repose la pointe AFM). Les points de transition $z = Z_0$ et $z = Z_1$ indiquent le début et la fin de l'interaction électrostatique, avant contact pointe-échantillon. Le points $z = Z_1$ indique aussi le début du contact déformable (modèle de Hertz). Dans le domaine δ, le point δ_2 est la limite entre le contact déformable (loi de Hertz) et le contact non déformable (loi de Hooke). (b) Sur une courbe de retrait relative à une bactérie, les sauts du signal indiquent que la pointe perd contact de façon similaire au détachement d'une bande auto-agrippante velcro, avec les protéines adhésives recouvrant le corps de la bactérie, jusqu'à une perte totale du contact (force nulle sur la partie gauche de la courbe). Les régions d'intérêt sont des intervalles $z \in [Z_j^{begin}, Z_j^{end}]$ dans lesquelles la courbe de force est une fonction décroissante de z. Plusieurs types de modèles paramétriques existent. Ils sont applicables dans chaque région d'intérêt.

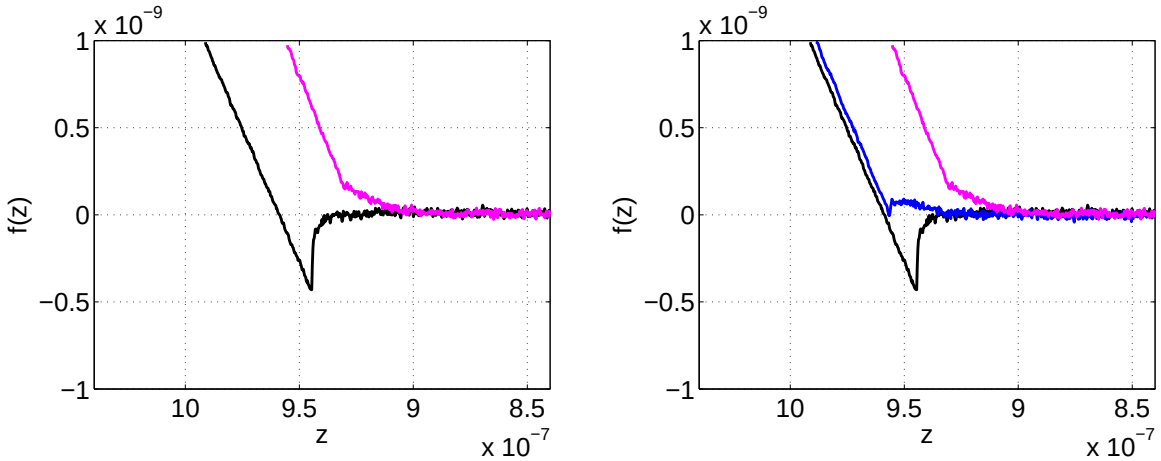


FIGURE 1.6 – Imagerie force-volume d'une nano-particule de goethite (en complément de la Fig. 1.3). (a) Acquisition de courbes de force mesurées soit en surface de la particule de goethite (interactions répulsives, force positive) soit sur le support (interactions attractives). (b) Trois courbes de force expérimentales (approche) correspondant à trois pixels de l'image. On observe deux courbes avec des interactions adhésives et répulsives, respectivement, et une troisième courbe (au milieu) où les deux motifs adhésif et répulsif apparaissent simultanément. Cette courbe correspond à un pixel à la frontière entre la nano-particule et la lame de verre. La courbe peut être modélisée comme le mélange de deux signaux élémentaires adhésif et répulsif, chacun étant retardé.

pour le i-ème pixel (x_i, y_i) est modélisée par :

$$f(x_i, y_i, z) = \sum_{k=1}^{p} a_{ik} s_k(z - z_{ik}) \tag{1.1}$$

où les signaux source s_k décrivent les forces d'interaction élémentaires, et chaque coefficient a_{ik} représente le poids de la k-ème source à l'intérieur du i-ème pixel. Les retards z_{ik}, homogènes à des distances pointe-échantillon, caractérisent la topographie de la k-ème source à l'intérieur du i-ème pixel. Le modèle (1.1) se réduit à un modèle linéaire instantané lorsque les retards z_{ik} sont nuls, c'est-à-dire lorsque la topographie de l'échantillon est plane.

Le modèle (1.1) est illustré par la Fig. 1.6 dans le cas de l'imagerie d'une nano-particule de goethite posée sur une lame de verre. Ici, deux sources sont présentes dans les mélanges. Un pixel situé sur la lame de verre conduit à une interaction attractive (forces négatives) alors qu'un pixel situé à l'intérieur de la nano-particule conduit à une interaction répulsive (forces positives). En un point frontière ou dans une zone où l'épaisseur de la nano-particule est faible, la courbe de force présente un mélange des deux motifs adhésif et répulsif car la pointe AFM est à la fois en interaction avec la nano-particule et avec la lame de verre. Dans ce mélange, les signaux élémentaires sont retardés à cause du changement de topographie de la nano-particule et de l'inclinaison de la lame de verre.

La validation du modèle de mélange avec retards (1.1) sur les données expérimentales présentées dans les Fig. 1.3 et 1.6 a été effectuée antérieurement au début de la thèse, et présentée dans la communication [5]. La validation du modèle, sur ce cas particulier, s'est appuyée sur la connaissance de la forme attendue des deux signaux sources $s_1(z)$ et $s_2(z)$. Cette procédure de validation consistait, pour chaque mélange i, à estimer les quatre paramètres a_{i1}, a_{i2}, z_{i1} et z_{i2} au sens des

moindres carrés, en minimisant

$$\sum_z [f(x_i, y_i, z) - a_{i1} s_1(z - z_{i1}) - a_{i2} s_2(z - z_{i2})]^2 \tag{1.2}$$

Ce traitement, répété sur tous les pixels i, nous a permis de vérifier que le modèle de mélange avec retard est bien pertinent pour les courbes d'approche car le résidu obtenu $f(x_i, y_i, z) - a_{i1} s_1(z - z_{i1}) - a_{i2} s_2(z - z_{i2})$ est faible et du même ordre de grandeur pour tous les pixels. En revanche, les courbes de retrait n'obéissent pas au modèle (1.1), car un phénomène d'hystérèse se produit lors de la mesure des courbes de force. D'autre part, les courbes d'approche mesurées pour les systèmes déformables comme les bactéries, n'obéissent pas parfaitement au modèle (1.1). En effet, les bactéries sont des micro-organismes vivants et l'acquisition répétée de courbes de force modifie parfois leur structure et engendre des variabilités sur les mesures des courbes. Le modèle de mélange (1.1) reste utilisable en première approximation.

Le traitement envisagé consiste finalement à rechercher des forces d'interaction élémentaires dans les données par une procédure de séparation de sources. Cette procédure doit identifier les sources (et déterminer leur nombre), estimer leurs poids a_{ik} et leurs retards z_{ik} dans chaque mélange. Pour les systèmes bactériens, les mélanges peuvent présenter plus de deux sources et leur forme n'est pas connue *a priori*. La procédure de séparation de sources doit s'appuyer sur les modèles paramétriques par morceaux qui décrivent les interactions entre la pointe AFM et un échantillon homogène. La recherche des signaux sources dans une image force-volume s'effectue à partir d'un grand nombre de mélanges car il y autant de mélanges que de pixels dans l'image.

Nous proposons à présent des formulations mathématiques pour les deux problèmes inverses introduits ci-dessus. Elles s'appuient sur la notion de modèles parcimonieux. Pour une courbe de force, la notion de parcimonie est liée au nombre de points de discontinuité dans le signal : points correspondant à des sauts du signal et/ou des changements de pente, des changements de courbure. Une représentation parcimonieuse consiste à détecter ces points de discontinuité et à approcher la courbe de force par un signal polynômial par morceaux.

1.2.2 Formulation du lissage d'un signal

Minimisation de critères penalisés

Ce traitement a pour but de lisser une courbe de force donnée [2] $z \mapsto f(z)$ tout en détectant les points de discontinuités dans le signal lissé (par exemple, les discontinuités d'ordres 0 et 1, c'est-à-dire les sauts du signal et les changements de pente). Nous utiliserons la terminologie de " lissage par morceaux ". Physiquement, ces points sont souvent liés à des grandeurs physiques caractéristiques, comme le point de contact pointe-échantillon ou la surface de contact (voir la Fig. 1.5). Pour pouvoir interpréter une courbe de force, une estimation précise de ces points de discontinuité est indispensable. En termes de traitement de signal, il s'agit de **reconstruire un signal $g(z)$ lisse par morceaux à partir d'une courbe de force expérimentale $f(z)$ avec un faible nombre de morceaux (parcimonie de $g'(z)$ ou $g''(z)$)**.

Il existe beaucoup d'approches pour traiter ce problème. Nous nous sommes focalisés sur des approches de type moindres carrés pénalisés. Pour fixer les idées, considérons la douceur à l'ordre 1, c'est-à-dire l'hypothèse que la dérivée première $g'(z)$ est faible presque partout [3]. Il s'agit de

2. Pour simplifier, nous omettons ici la dépendance de f par rapport à x_i et y_i, car la position du pixel (x_i, y_i) est fixe.

3. ou nulle presque partout dans le cas où g' est un signal parcimonieux.

construire et de minimiser un critère du type :

$$\mathcal{J}(g) = \sum_z [f(z) - g(z)]^2 + \lambda \sum_z \phi(g'(z)) \tag{1.3}$$

Ici, g est le signal " lissé par morceaux " à reconstruire, $\phi : \mathbb{R} \to \mathbb{R}_+$ est une fonction paire et croissante sur $\mathbb{R}_+$, et le terme de pénalisation $\sum_z \phi(g'(z))$ a pour rôle de contraindre le signal $g(z)$ à être continu par morceaux. On distingue trois types de fonctions ϕ.

- Lorsque ϕ est choisie strictement convexe et dérivable, on reconstruit un signal $g(z;\lambda)$ continu par morceaux, mais avec des discontinuités peu franches ; en d'autres termes, $z \mapsto g'(z;\lambda)$ n'est pas un signal parcimonieux. Du point de vue optimisation, $\mathcal{J}$ est strictement convexe, on peut donc mener son optimisation par un algorithme de descente classique.
- Le choix d'une fonction ϕ non convexe, comme la fonction quadratique tronquée, a pour but de favoriser davantage la reconstruction de signaux comportant des discontinuités. Dans ce cas, le critère $\mathcal{J}$ n'est en général ni convexe ni monomodal, et il faut recourir à des algorithmes d'optimisation spécifiques pour trouver son minimum global [6].
- Un compromis entre les stratégies précédentes est le choix d'une fonction convexe au sens large, comme la fonction $\phi(t) = |t|$. Ce choix permet de reconstruire un signal continu par morceaux, et rend le critère $\mathcal{J}$ strictement convexe, donc monomodal. Cependant, $\mathcal{J}$ est non différentiable, il faut dont recourir à des algorithmes spécifiques de minimisation [7], [8].

Nous avons traité par cette approche plusieurs jeux de données expérimentales. Les fonctions ϕ strictement convexes [9] donnent des résultats satisfaisants, mais le signal reconstruit n'est pas à dérivée parcimonieuse, et, de plus, ces méthodes imposent le réglage empirique de plusieurs hyper-paramètres. Les résultats obtenus avec des fonctions ϕ non convexes couplées avec un algorithme d'optimisation local sont parfois très convaincants, mais restent conditionnés par la solution initiale choisie pour l'algorithme d'optimisation [6]. Les algorithmes d'optimisation liés au choix $\phi(t) = |t|$ sont particulierèment intéressants, car il n'y a qu'un seul hyperparamètre à régler (λ). De plus, il existe des algorithmes qui fournissent le continuum des solutions $g(z;\lambda)$ en fonction de λ [8], [10].

Utilisation de la pseudo-norme ℓ_0

Pour contraindre la dérivée première du signal g à être parcimonieuse, c'est-à-dire non nulle pour un nombre limité de valeurs de z seulement, il faut que ϕ soit non différentiable en 0 [11]. C'est par exemple le cas de la fonction valeur absolue (le critère $\mathcal{J}$ est dit ℓ_2-ℓ_1). Cette fonction présente néanmoins l'inconvénient de pénaliser davantage les fortes valeurs de $g'(z)$. Choisir une fonction ϕ non convexe permet de pallier ce problème. Nous avons choisi d'utiliser le coût ℓ_0 (ou pseudo-norme ℓ_0). C'est la fonction binaire définie par :

$$\phi(t) = |t|_0 = \begin{cases} 0, & \text{si } t = 0\,; \\ 1, & \text{si } t \neq 0. \end{cases} \tag{1.4}$$

Nous nommons alors le critère $\mathcal{J}$ "critère ℓ_2-ℓ_0", car il est composé d'un terme quadratique et d'un terme lié à la pseudo-norme ℓ_0 :

$$\mathcal{J}(g) = \sum_z [f(z) - g(z)]^2 + \lambda \sum_z |g'(z)|_0 \tag{1.5}$$

Comme la fonction $t \mapsto |t|_0$ est binaire, la minimisation de (1.5) est un **problème d'optimisation combinatoire** qui repose sur l'activation ou non d'une discontinuité en chaque échantillon $f(z)$. Il s'agit de décider pour tout z, si $g'(z) \neq 0$ ou non.

Ce type de problème d'optimisation est connu pour être NP-difficile [12] : il n'existe pas d'algorithme optimal, hormis celui qui consiste à faire une recherche exhaustive des points de disconti-

nuité. Néanmoins des algorithmes sous-optimaux efficaces existent, comme par exemple, ceux qui reposent sur la modélisation des échantillons du signal g par un processus Bernoulli-Gaussien [13], [14]. D'autres algorithmes, de type ajout-retrait (forward-backward), ont été développés dans le domaine de la régression par sélection de variables [15]. Dans cette thèse, nous faisons le lien entre ces deux familles d'algorithmes. De plus, nous développons des algorithmes sous-optimaux permettant d'obtenir le continuum des solutions $\lambda \mapsto g(z;\lambda)$ pour le critère ℓ_2-ℓ_0, en tant qu'extension des algorithmes liés aux critères ℓ_2-ℓ_1. Il s'agira de comparer les performances de cet algorithme aux algorithmes existants pour résoudre le problème ℓ_2-ℓ_0 (qui travaillent à λ fixé) [14], et d'étudier des heuristiques permettant de choisir judicieusement la valeur de λ.

Détection conjointe de discontinuités d'ordres différents

La formulation (1.5) consiste à approcher un signal par un signal constant par morceaux, car la dérivée première du signal lissé g est parcimonieuse (nulle presque partout). Par extension, on peut désirer faire une approximation par un polynôme de degré p par morceaux (un signal affine par morceaux pour $p = 1$, un signal quadratique par morceaux pour $p = 2$) en remplaçant le critère (1.5) par :

$$\mathcal{J}(g) = \sum_z [f(z) - g(z)]^2 + \lambda \sum_z |g^{(p+1)}(z)|_0 \tag{1.6}$$

En effet, minimiser (1.6), c'est contraindre la dérivée d'ordre $p+1$ du signal g à être parcimonieuse, c'est-à-dire non nulle pour un nombre limité de valeurs de z seulement. Le paramètre λ règle le compromis entre le degré de parcimonie (le nombre de morceaux) et la qualité de l'approximation de f par g. Pour une grande valeur de λ, l'erreur d'approximation est grande mais le nombre de points de discontinuité est faible. Des approximations de meilleure qualité sont obtenues pour des faibles valeurs de λ, au prix d'un plus grand nombre de discontinuités.

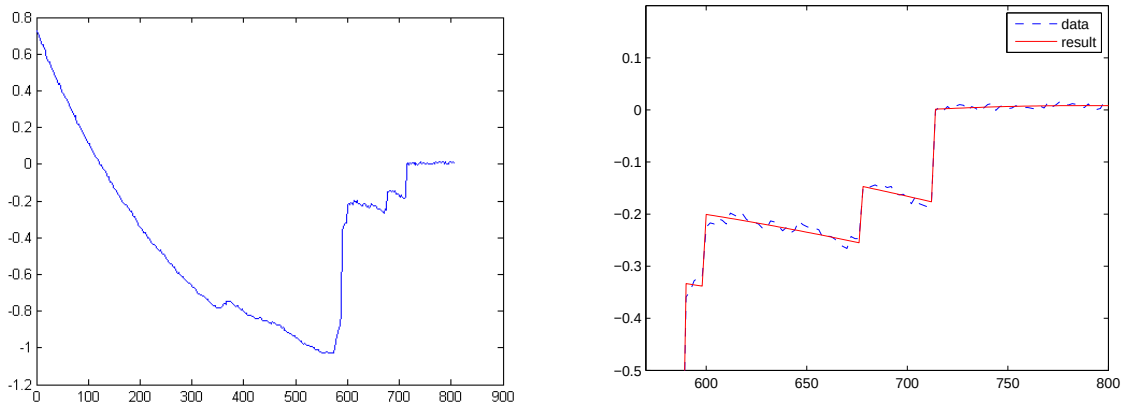
Si le signal g obtenu en minimisant (1.6) est un polynôme de degré p par morceaux, aucune contrainte n'est imposée quant au raccordement de deux morceaux consécutifs (comme la continuité ou le fait que les dérivées à droite et à gauche soient égales en un point frontière entre deux morceaux). Or, il est parfois souhaitable d'imposer ce type de contrainte (voir par exemple les modèles de la Fig. 1.5 (a) où deux morceaux consécutifs sont raccordés continûment). Pour détecter des points de discontinuité **à plusieurs ordres simultanément** dans un signal, on remplace le critère (1.6) par :

$$\mathcal{J}(g_0, g_1, \ldots, g_{P-1}) = \sum_z \left[f(z) - \sum_{p=0}^{P-1} g_p(z) \right]^2 + \lambda \sum_{p=0}^{P-1} \sum_z |g_p^{(p+1)}(z)|_0 \tag{1.7}$$

Le signal lissé par morceaux s'exprime donc sous la forme $g = \sum_p g_p$, où chaque signal g_p est un polynôme de degré p par morceaux (g_0 est constant par morceaux, g_1 est affine par morceaux, *etc.*). Les points de discontinuité d'ordre p sont les points en lesquels la dérivée d'ordre $(p+1)$ de g_p est non nulle. Le lissage polynômial par morceaux est illustré sur la Fig. 1.7. P vaut 3, le signal lissé par morceaux s'exprime donc sous la forme $g = g_0 + g_1 + g_2$, où $g_0', g_1^{(2)}$ et $g_2^{(3)}$ sont parcimonieux. Dans ce cas, la fonction g n'est rien d'autre qu'une fonction quadratique par morceaux.

Nous étudierons deux stratégies pour la recherche des points de discontinuité, appelées **sélection de variables scalaires et vectorielles** :

1. Pour la sélection de variables scalaires, les discontinuités détectées à des ordres différents ne sont généralement pas positionnées en une même position z. Une " variable scalaire " est définie comme un couple (z, p) qui décrit la position z d'une discontinuité et son ordre p ;
2. Il est parfois avantageux d'imposer l'apparition de discontinuités à *tous les ordres* $p = 0, 1,$ *etc.* en une même position z, lorsque cette position est activée. C'est le cas lorsqu'on désire



(a) (b)

FIGURE 1.7 – Traitement d'une courbe de force relative à des levures de *Candida Albicans* [16]. (a) Courbe de force expérimentale (courbe de retrait) ; (b) Affichage du zoom de (a) (dernière partie) et du signal lissé, continu par morceaux, obtenu à partir de (a) : lissage du signal par un signal quadratique par morceaux, par application de l'algorithme SBR pour la recherche des morceaux.

reconstruire un signal polynômial par morceaux avec un nombre minimal de morceaux. Ainsi, l'activation de discontinuités à tous les ordres en une même position z est associée à une " variable vectorielle " $\{g_p^{(p+1)}(z), \forall p\} \neq \mathbf{0}$.

En résumé, la sélection de variables scalaires favorise le raccordement entre deux morceaux consécutifs du signal $g(z)$ (raccordement continu, raccordement des dérivées d'ordre p) alors que pour la sélection de variables vectorielles, deux morceaux consécutifs ne sont pas raccordés et les dérivées d'ordre $p \neq p_0$ à gauche et à droite d'un point de discontinuité d'ordre p_0 ne sont généralement pas égales.

1.2.3 Séparation de sources parcimonieuses retardées

On modélise la courbe de force $f(x_i, y_i, z)$ mesurée en un pixel (x_i, y_i) comme le mélange de p interactions élémentaires (ou *sources*) s_k selon le modèle de mélange avec retards (1.1). La recherche conjointe des sources, de leur topographie z_{ik} et des coefficients du mélange a_{ik} à partir d'une image force-volume $f(x, y, z)$ est un problème de séparation de sources. C'est un problème classique du traitement du signal, mais qui, compte tenu des retards z_{ik}, est loin d'être trivial. La difficulté du problème est liée au fait que la topographie de l'échantillon est inconnue, en plus des sources et de leurs poids. Le problème est connu pour posséder de nombreuses indéterminations. En particulier, les sources s_k ne peuvent être séparées qu'à un facteur d'échelle près λ_k et qu'à un retard près z'_k :

$$\begin{aligned} (\lambda_k a_{ik}) s_k &= a_{ik}(\lambda_k s_k), \\ s_k(z - z_{ik}) &= s'_k(z - z_{ik} + z'_k), \quad \text{où} \quad s'_k(z) \triangleq s_k(z - z'_k) \end{aligned}$$

Il est essentiel d'imposer des contraintes pour pouvoir assurer l'identifiabilité du mélange (1.1). Typiquement, les sources sont supposées blanches et indépendantes [17], ou leur transformée de Fourier est supposée parcimonieuse [18]. Malheureusement, ces hypothèses sont incompatibles avec la forme attendue des signaux sources en spectroscopie de force. Des hypothèses plus réalistes reposent sur :

1. les modèles physiques qui décrivent la force d'interaction entre une pointe et un échantillon homogène, que ce soit pour la phase d'approche (forces de van der Waals, électrostatiques, élastiques, *etc.*) ou de retrait (forces d'adhésion, de capillarité, de liaison, *etc.*). Pour chaque

type d'interaction, il existe des modèles. La force d'interaction peut alors s'exprimer comme une fonction paramétrique $s_k(z;\boldsymbol{\theta}_k)$, où le vecteur $\boldsymbol{\theta}_k$ regroupe l'ensemble des paramètres caractéristiques de la forme de la courbe de force (par exemple une partie exponentielle, une discontinuité) [19], [5]. Ce sont ces signaux paramétriques qu'il s'agit d'extraire d'une image force-volume par une procédure de décomposition [19], [20] ;

2. une représentation parcimonieuse de chaque source $s_k(z)$ par un signal polynômial par morceaux (voir cidessus). Chaque signal source est alors décrit par chacun de ses points de discontinuité d'ordre 0, 1, *etc.* La procédure de séparation de sources consiste finalement à reconstruire les signaux sources qui soient, pour un degré de parcimonie donné, le plus en adéquation avec les mélanges.

Compte tenu du développement d'algorithmes approximation parcimonieuse, nous avons opté pour la deuxième approche. En pratique, chaque source est décrite par un polynôme par morceaux de degré 1 ou 2. Le principe de l'algorithme de séparation est de détecter les points de discontinuité d'ordre 0 et 1 (voire d'ordre 2) dans chaque courbe de force $z \mapsto f(x_i, y_i, z)$ en appliquant l'algorithme de lissage par morceaux, puis de mettre en correspondance les différentes représentations parcimonieuses. Cela permet, par recoupement, d'estimer le nombre de sources et de décrire chaque source $s_k(z)$ par ses points de discontinuité (position z de la discontinuité et ordre p).

1.3 Contributions principales

Dans cette partie, nous résumons les contributions principales de la thèse et nous les illustrons par quelques résultats de traitement de données réelles.

1.3.1 Approximation parcimonieuse par minimisation d'un critère mixte ℓ_2-ℓ_0

plutôt que de développer des algorithmes spécifiques à la minimisation des critères (1.5), (1.6) et (1.7), nous avons opté pour un cadre plus général : l'approximation d'un signal à partir d'un nombre limité d'éléments d'un dictionnaire [21]. Nous verrons au paragraphe III-C que la minimisation des critères (1.5), (1.6) et (1.7) se ramène facilement a ce cadre, où le dictionnaire est une matrice de Tœplitz ou Tœplitz par bloc qui représente un ou des opérateurs d'intégration numérique.

Nous avons développé des algorithmes heuristiques (dans le sens où la convergence vers la solution optimale n'est pas garantie) permettant d'optimiser des critères mixtes " ℓ_2-ℓ_0 " du type

$$\min_{\{\boldsymbol{x}, \|\boldsymbol{x}\|_0 \leqslant k\}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 \tag{1.8}$$

où $\|\boldsymbol{x}\|_0$ est le coût ℓ_0 de $\boldsymbol{x}$, défini comme le nombre d'éléments non nuls de $\boldsymbol{x}$, et l'entier k désigne le degré de parcimonie. La formulation (1.8) permet d'approcher un signal $\boldsymbol{y}$ à partir d'au plus k colonnes de la matrice $\boldsymbol{A}$. Les **nouveaux algorithmes d'approximation parcimonieuse** que nous avons conçu reposent sur la minimisation du critère auxiliaire :

$$\mathcal{J}(\boldsymbol{x};\lambda) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{x}\|_0 \tag{1.9}$$

Leur principe repose sur le fait que, contrairement à la forme contrainte (1.8), la minimisation du critère pénalisé (1.9) permet d'envisager non seulement l'ajout mais aussi le retrait de variables actives dans un algorithme de descente déterministe de type " ajout-retrait " (forward-backward en anglais).

L'algorithme de minimisation du critère (1.9) à λ fixé, est nommé *Single Best Replacement* (SBR). Il est inspiré de l'algorithme *Single Most Likely Replacement* (SMLR), initialement proposé pour résoudre des problèmes de déconvolution de signaux impulsionnels [13], [14], [22]. SBR est

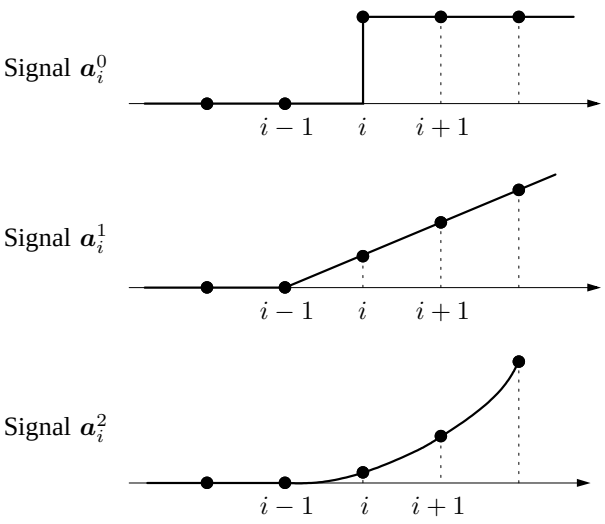


FIGURE 1.8 – Signaux $\boldsymbol{a}_i^p$ liés à l'activation d'une discontinuité d'ordre p en position i. $\boldsymbol{a}_i^0$ est la fonction de Heaviside discrète, $\boldsymbol{a}_i^1$ est la fonction rampe, et $\boldsymbol{a}_i^2$ est la fonction quadratique positive. Chaque signal vaut 1 pour la position i et son support est égal à $\{i, \ldots, m\}$.

un algorithme itératif qui consiste à ajouter ou à retirer une seule colonne de $\boldsymbol{A}$ à l'ensemble des colonnes sélectionnées entre une itération et l'itération suivante. Plus précisément, l'algorithme recherche à chaque itération, le remplacement élémentaire (ajout ou retrait d'une colonne) qui permet de faire décroître le plus possible la valeur du critère $\mathcal{J}(\boldsymbol{x}; \lambda)$.

1.3.2 Algorithme de continuation

La difficulté posée par l'utilisation de l'algorithme SBR réside dans le fait que l'hyperparamètre λ doit être réglé empiriquement, et correspond à un niveau de parcimonie fixé. Nous avons développé une version étendue de SBR, qui permet de fournir des solutions à différents niveaux de parcimonie $\lambda_1 > \lambda_2 > \ldots > \lambda_J$. Chaque solution $\boldsymbol{x}(\lambda_j)$ s'obtient récursivement à partir de la solution précédente $\boldsymbol{x}(\lambda_{j-1})$ en utilisant l'algorithme SBR, et les valeurs de λ_j sont calculées de façon adaptative aux données. Le nouvel algorithme est appelé *Continuation SBR* (CSBR). Son principal intérêt est qu'une estimation (sous-optimale) de toutes les solutions $\{\boldsymbol{x}(\lambda), \forall \lambda \geqslant 0\}$ se déduit directement des J solutions $\boldsymbol{x}(\lambda_j), j = 1, \ldots, J$ par un procédé de " continuation ". L'algorithme CSBR permet également, au prix d'une modification mineure, d'estimer un ensemble de solutions du problème contraint (1.8) pour tout k.

Nous avons comparé les performances de CSBR à celles des algorithmes existants pour optimiser des critères mixtes "ℓ_2-ℓ_0" [23],[24],[25]. CSBR fournit des résultats au moins aussi bons au prix d'une complexité (temps de calcul) supérieure, mais du même ordre de grandeur.

1.3.3 Détection conjointe de discontinuités

Comme nous l'avons vu dans la partie II, la détection de discontinuités peut être formulée comme l'optimisation d'un critère mixte ℓ_2-ℓ_0, car il s'agit d'approcher au mieux un signal donné par un signal lisse par morceaux sous la contrainte que le nombre de points de discontinuité (ou de morceaux) soit limité. Nous montrons à présent que ces problèmes peuvent aussi être formulés sous la forme (1.8) ou (1.9). La formulation ci-dessous est entièrement discrète, c'est-à-dire que le signal $f(z)$ est représenté par un vecteur $\boldsymbol{y} = [y_1, \ldots, y_m]^t \in \mathbb{R}^m$ et la position d'un point de discontinuité est un indice $i \in \{1, \ldots, m\}$.

Construction du dictionnaire

Nous posons le problème de la détection conjointe des discontinuités d'ordres $p = 0, 1, \ldots, P-1$ dans un signal $\boldsymbol{y}$ comme un problème de sélection de variables, où une variable est associée à une discontinuité (position i et ordre p). Le dictionnaire est construit de façon à ce que tous les types de discontinuité recherchés puissent être présents pour une position donnée. Ainsi, pour les discontinuités d'ordre p, nous considérons la matrice de Tœplitz $\boldsymbol{A}_p$ dont les colonnes résultent de l'échantillonnage sur une grille régulière d'un signal polynômial d'ordre p, en envisageant tous les points de discontinuité possibles :

$$\boldsymbol{A}_p = \begin{bmatrix} 1^p & 0 & \ldots & 0 \\ 2^p & 1^p & & \vdots \\ 3^p & 2^p & & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \vdots & \vdots & \ldots & 1^p \\ m^p & (m-1)^p & \ldots & 2^p \end{bmatrix}$$

La Fig. 1.8 illustre cette définition dans les cas $p = 0, 1$ et 2. La i-ème colonne de la matrice $\boldsymbol{A}_0$ est une fonction de Heaviside discrète, la i-ème colonne de la matrice $\boldsymbol{A}_1$ est une fonction rampe, la i-ème colonne de la matrice $\boldsymbol{A}_2$ et une fonction quadratique positive, *etc.*

Le dictionnaire permettant la détection conjointe des discontinuités d'ordres $p = 0, 1, \ldots, P-1$ est finalement obtenu en concaténant l'ensemble des dictionnaires :

$$\boldsymbol{A} = [\boldsymbol{A}_0 | \boldsymbol{A}_1 | \ldots | \boldsymbol{A}_{P-1}] \tag{1.10}$$

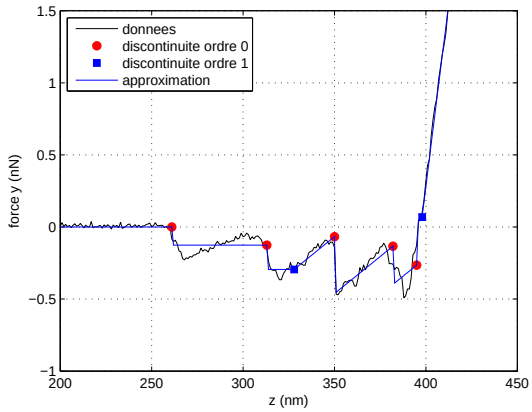
et le lissage d'un signal polynômial par morceaux (1.7) se récrit finalement comme l'approximation de $\boldsymbol{y}$ par $\boldsymbol{A}\boldsymbol{x}$. Les poids $\boldsymbol{x}_j$ représentent ici l'amplitude des discontinuités (analogue à $g_p^{(p+1)}(z)$ dans (1.7)) et les indices j pour lesquels $x_j \neq 0$ représentent à la fois la position z de la discontinuité et son ordre p.

Dans les deux paragraphes suivants, nous proposons deux formulations de l'approximation d'un signal $\boldsymbol{y}$ par le biais du dictionnaire $\boldsymbol{A}$, vue comme des problèmes de sélection de variables.

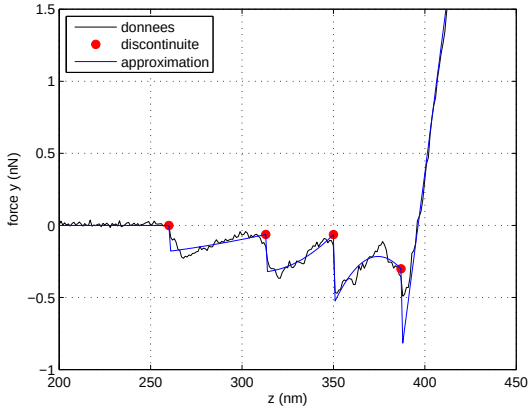
Sélection de variables scalaires

La sélection de variables scalaires est formulée comme la minimisation de $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ sous la contrainte $\|\boldsymbol{x}\|_0 \leqslant k$, où $\|\boldsymbol{x}\|_0$ est le nombre total de discontinuités, et l'entier k contrôle le nombre maximal de discontinuités.

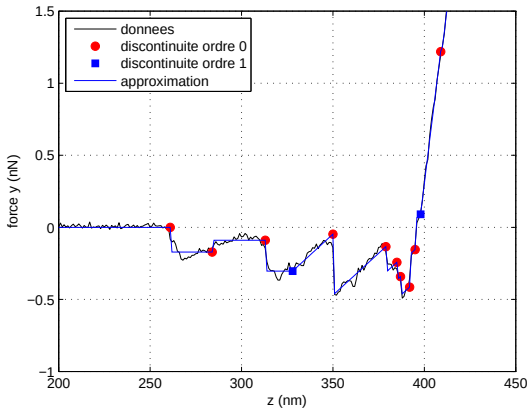
Pour traiter ce problème, nous avons appliqué les algorithmes SBR et CSBR. L'algorithme CSBR fournit d'abord des approximations grossières avec peu de points de discontinuité (grandes valeurs de λ ou faibles valeurs de k), puis finalement des approximations plus fines avec davantage de points de discontinuité. Le critère d'arrêt retenu pour choisir une solution parmi cet ensemble de solutions est le suivant. On considère que les principales discontinuités ont été trouvées à l'itération k lorsque l'erreur d'approximation $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ est inférieure à un seuil. La valeur du seuil est réglée empiriquement en fonction de la variance du bruit de mesure, elle-même estimée à partir des données expérimentales [26].



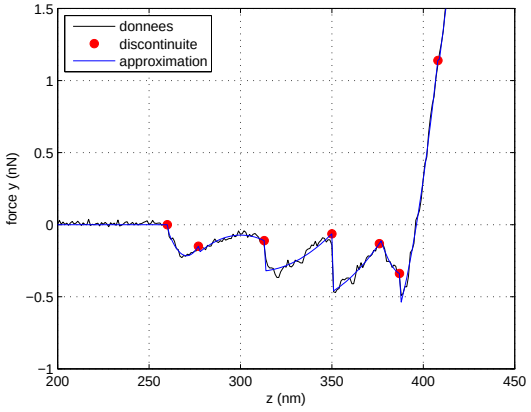
(a) $S = 7m\sigma^2$



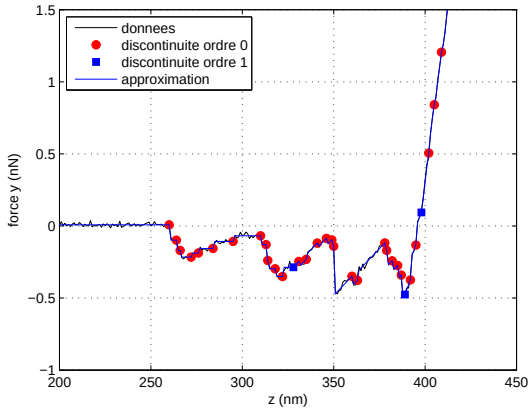
(d) $S = 7m\sigma^2$



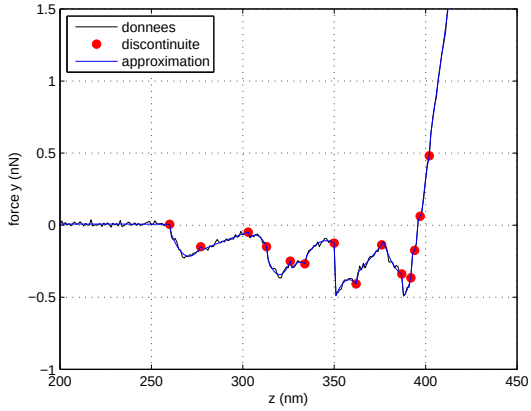
(b) $S = 3m\sigma^2$



(e) $S = 3m\sigma^2$



(c) $S = m\sigma^2$



(f) $S = m\sigma^2$

FIGURE 1.9 – Lissage par morceaux d'une courbe de force par sélection de variables scalaire et vectorielles. Le signal lissé est quadratique par morceaux, c'est-à-dire qu'on recherche des discontinuités aux ordres $p = 0$, 1 et 2. La courbe mesurée $y(z)$ est affichée en noir, et son approximation en bleu. (a-c) Approximations de la courbe de force par sélection de variables scalaires, obtenues avec différentes valeurs de seuil sur le résidu de l'approximation. Ces valeurs sont fixées à $7m\sigma^2$, $3m\sigma^2$, et $m\sigma^2$, respectivement, où m désigne le nombre d'échantillons dans le signal mesuré, et σ^2 est une estimation empirique de la variance du bruit. On représente par une couleur différente les discontinuités aux ordres 0 et 1. Aucune discontinuité à l'ordre 2 n'est trouvée. (d-f) Approximations de la courbe de force par sélection de variables vectorielles, avec les trois mêmes valeurs du seuil S. Les positions des discontinuités sont représentées d'une seule couleur, car des discontinuités à tous les ordres sont présentes simultanément.

Sélection de variables vectorielles

Dans l'approche précédente, le terme $\|\boldsymbol{x}\|_0$ compte le nombre de discontinuités pour chaque ordre et pour chaque position. La détection de discontinuités à des ordres différents et *pour une même position* (par exemple, un saut et un changement de dérivée dans le signal au même moment) nécessite plusieurs variables, puisqu'il faut sélectionner plusieurs colonnes de la matrice $\boldsymbol{A}$. Lorsqu'on désire reconstruire un signal polynômial par morceaux avec un nombre minimal de morceaux, il est avantageux d'autoriser l'apparition de discontinuités à tous les ordres $p = 0, 1, \ldots, P-1$ en une même position, en comptant 1 (soit une *position* de discontinuité) au lieu de P (P variables). Ainsi, une discontinuité en une position i est associée à une "variable vectorielle" $\boldsymbol{x}_i$, qui regroupe les amplitudes des P discontinuités relatives à la position i, pour un coût unitaire.

Pour formuler mathématiquement la sélection de variables vectorielles, on réordonne le vecteur $\boldsymbol{x}$ comme la concaténation des vecteurs $\boldsymbol{x}_i$ de taille P : $\boldsymbol{x}^t = [\boldsymbol{x}_1^t, \ldots, \boldsymbol{x}_m^t]$. Le décompte du nombre de positions i sélectionnées s'écrit

$$\delta_P(\boldsymbol{x}) \triangleq \sum_{i=1}^{n} \mathbb{1}_P(\boldsymbol{x}_i)$$

où $\mathbb{1}_P(\boldsymbol{x}_i)$ vaut 1 si $\boldsymbol{x}_i \neq \boldsymbol{0}$, et 0 sinon. La sélection de variables vectorielles se réduit à la minimisation de $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ sous la contrainte $\delta_P(\boldsymbol{x}) \leqslant$, où k contrôle le nombre maximal de points de discontinuité autorisés. Il s'agit d'autoriser la sélection d'au plus k variables vectorielles, soit kP variables scalaires.

L'algorithme d'optimisation utilisé est une simple adaptation de l'algorithme SBR. Il consiste à minimiser le critère pénalisé $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \lambda\delta_P(\boldsymbol{x})$ sur $\mathbb{R}^n$. À chaque itération, il s'agit toujours de tester chaque ajout et chaque retrait d'une variable (vectorielle), c'est-à-dire l'ajout de P discontinuités en une position i ($\boldsymbol{x}_i \neq \boldsymbol{0}$) ou, si la position i est active, le retrait des P discontinuités présentes ($\boldsymbol{x}_i = \boldsymbol{0}$). De même, l'algorithme de continuation CSBR s'adapte sans problème à la sélection de variables vectorielles.

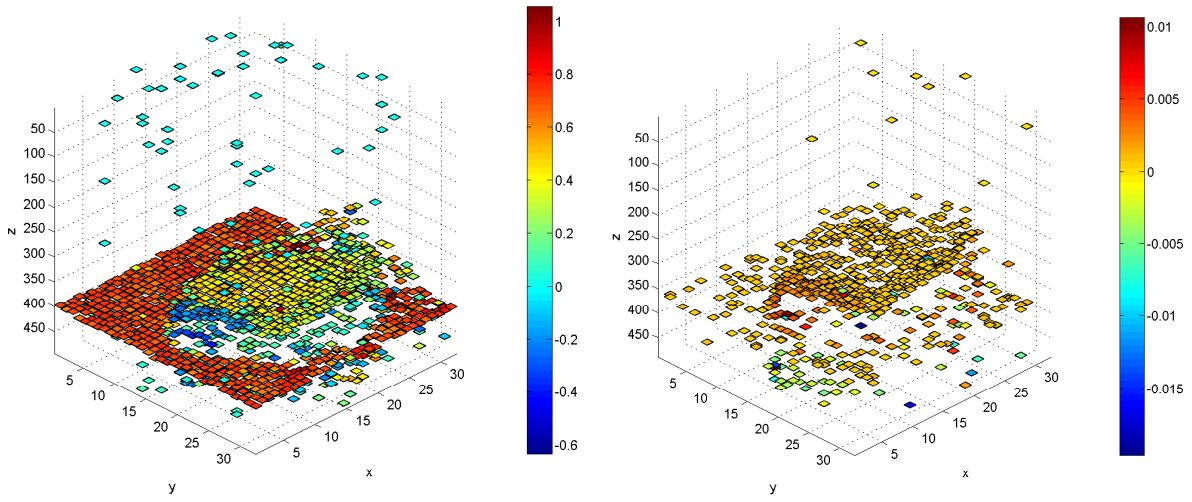
Traitement de données réelles

La Fig. 1.9 présente les résultats comparatifs des deux méthodes de sélection de variables (résultats extraits de [26]). Dans les deux cas, nous recherchons simultanément les discontinuités d'ordre 0, 1 et 2 dans le signal, ce qui revient à l'approcher par un polynôme de degré 2 par morceaux. La version vectorielle donne les résultats les plus convaincants en termes de résidu d'approximation $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ mais au prix de la sélection d'un plus grand nombre de colonnes. Par exemple, pour la sous-figure (d), 4 points de discontinuités sont détectés, c'est-à-dire 12 colonnes de la matrice $\boldsymbol{A}$ contre 8 points de discontinuité et 8 colonnes pour la sous-figure (a).

1.3.4 Séparation de sources parcimonieuses retardées

Nous proposons une solution au problème de la séparation de sources parcimonieuses retardées, où chaque source est décrite par la position de ses discontinuités d'ordre $p = 0, 1$, *etc.* et par l'amplitude de chaque discontinuité. La séparation de sources à partir des données brutes peut se résumer schématiquement par les deux traitements :

1. Lissage de chaque courbe de force $f(x_i, y_i, z)$ et estimation de ses discontinuités d'ordre 0, 1 et 2. Ce jeu de paramètres (pour chaque discontinuité : position, amplitude, et ordre de la dérivée) est noté $\boldsymbol{\theta}_i$.
2. Séparation des sources en mettant en correspondance les discontinuités $\boldsymbol{\theta}_i$ des différentes courbes (pour tous les pixels i).



(a) Discontinuités d'ordre 1 (b) Discontinuités d'ordre 2

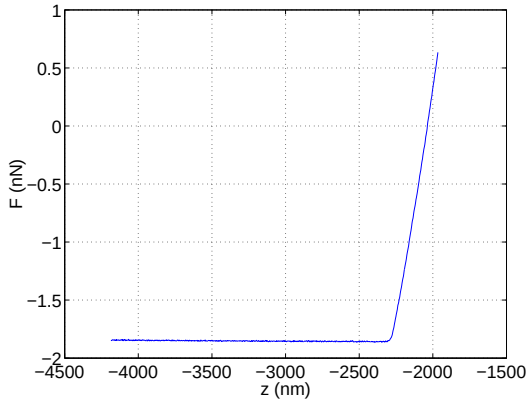
FIGURE 1.10 – Visualisation d'une image force-volume : représentation 3D des discontinuités $g_p^{(p+1)}(x_i, y_i, z) \neq 0$ estimées pour chacune des courbes de force $z \mapsto f(x_i, y_i, z)$, pour les ordres $p = 1$ et 2. Ces images 3D sont relatives à une cellule de staphylocoque doré. Les couleurs sont liées aux valeurs des amplitudes $g_p^{(p+1)}(x_i, y_i, z)$. Seules les amplitudes non nulles sont représentées.

Du point de vue théorique, nous discuterons des conditions sous lesquelles la solution du problème de séparation de mélange avec retards (1.1) est unique dans un cadre non bruité. Du point de vue pratique, le traitement 1) consiste à appliquer l'algorithme CSBR autant de fois que de mélanges. Chacune des courbes lissées par morceaux s'écrit $g(x_i, y_i, z) = \sum_p g_p(x_i, y_i, z)$ conformément au modèle (1.7). À p fixé, l'image $g_p^{(p+1)}(x_i, y_i, z)$ qui réunit les représentations parcimonieuses de toutes les courbes de force lissées à l'ordre p est visualisable en 3D. Pour cela, on représente pour chaque pixel (x_i, y_i) et pour chaque position de discontinuité z, l'amplitude $g_p^{(p+1)}(x_i, y_i, z)$ de la discontinuité par une couleur. Les représentations 3D obtenues pour une cellule de staphylocoque doré sont affichées sur la Fig. 1.10 pour $p = 1$ et $p = 2$. Sur ces figures, on retrouve notamment la topographie attendue de la cellule (sous-figure (a)) et des informations liées au contact mécanique (sous-figure (b)). Ces images ne sont néanmoins pas totalement interprétables ; ce sont des résultats intermédiaires en vue de l'estimation de paramètres physiques ou de la séparation de signaux sources.

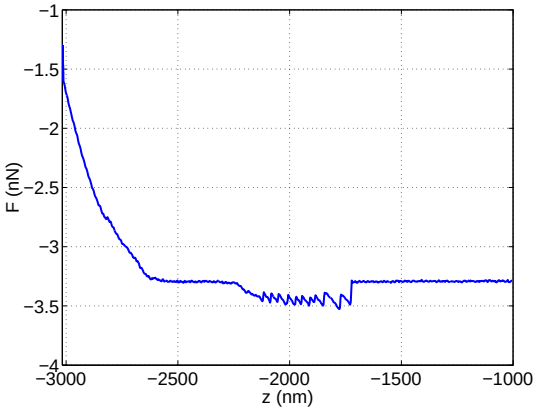
La séparation de sources se résume en un problème d'appariement des discontinuités. C'est un problème de classification qui consiste à étiqueter chaque discontinuité en l'associant à l'une ou l'autre des sources. La classification peut être effectuée en remarquant que la distance δz entre deux discontinuités associées à une source donnée est constante quelquesoit le mélange où la source est présente. De même, le rapport de leurs amplitudes est constant quelquesoit le mélange où la source est présente. La prise en compte de ces informations permet, au moins dans le cas non bruité, de résoudre l'appariement de discontinuités si une source possède au moins deux discontinuités. Des adaptations seront nécessaires si une ou plusieurs sources ne possède qu'une ou zéro discontinuité.

1.3.5 Segmentation d'une courbe de force et estimation de paramètres physiques

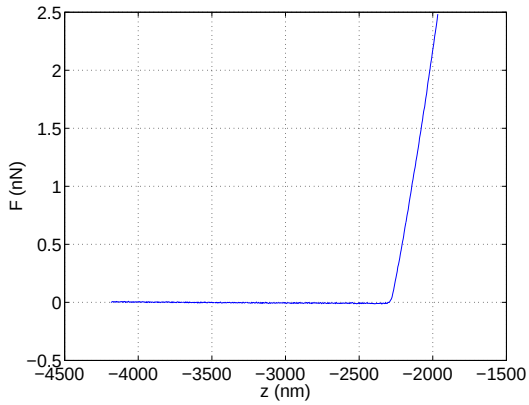
Une partie importante de cette thèse a été consacrée à la conception d'un outil de traitement pseudo-automatique d'une courbe de force, en collaboration avec le LCPME. Cet outil consiste en trois étapes :



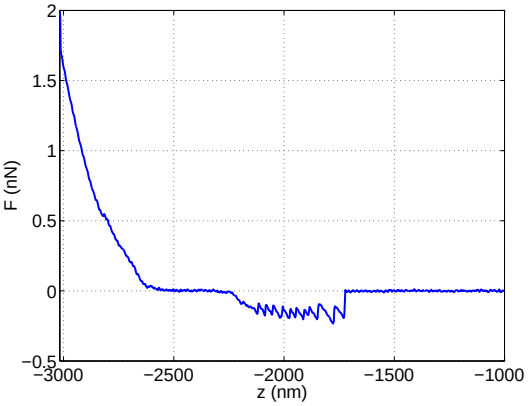
(a) Données brutes (approche)



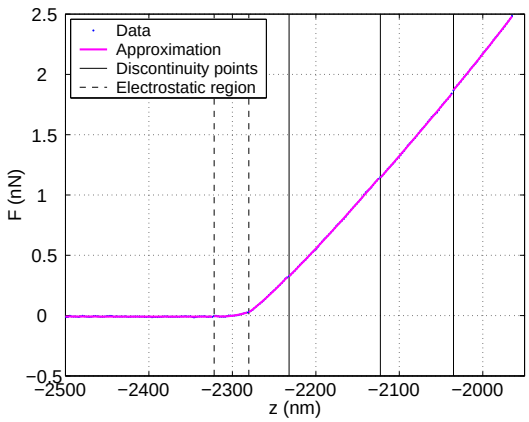
(d) Données brutes (retrait)



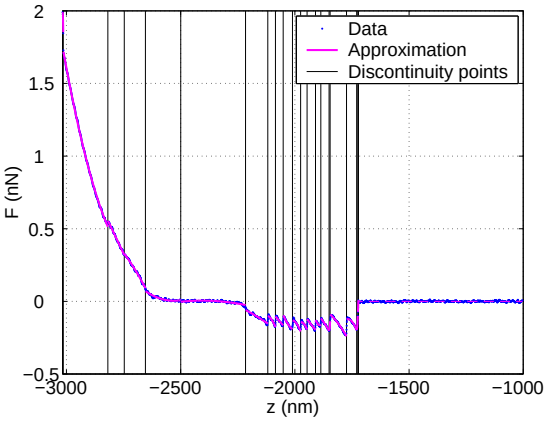
(b) Courbe prétraitée



(e) Courbe prétraitée

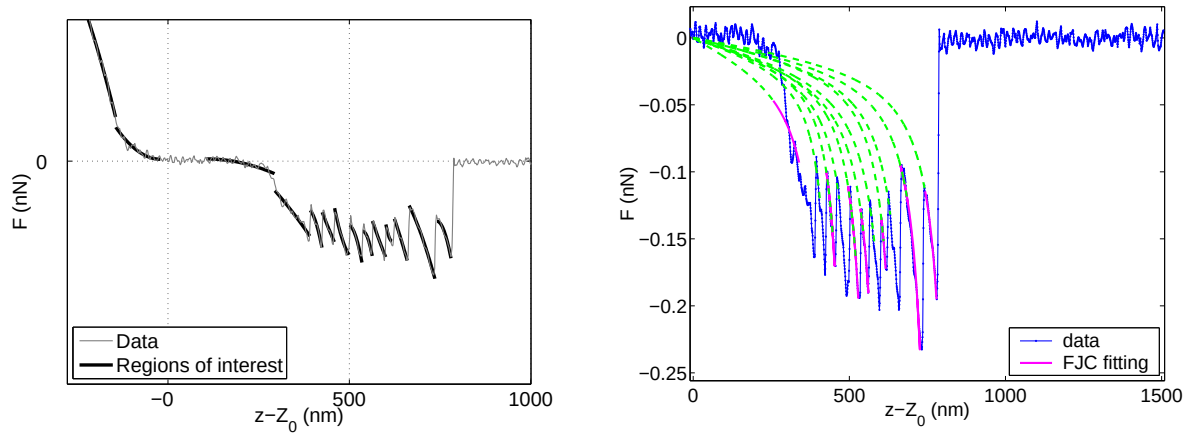


(c) Résultat de segmentation



(f) Résultat de segmentation

FIGURE 1.11 – Prétraitement d'une courbe de force et résultats de segmentation. (a-c) Courbe d'approche : données réelles, courbe prétraitée (soustraction d'une ligne de base affine), résultat de segmentation avec 6 points de discontinuité. Dans le zoom de la figure (c), seules 5 points de discontinuité sont visibles parmi les 6. Les deux points de discontinuité représentés en lignes pointillées sont les limites Z_0 et Z_1 de la région électrostatique. (d-f) Courbe de retrait. Le degré du polynôme est égal à $P = 2$, 20 points de discontinuité sont détectés.



(a) Détection de régions d'intérêt (b) Ajustement du modèle FJC

FIGURE 1.12 – Ajustement de modèles physiques sur une courbe de retrait : détection des régions d'intérêt (a) et ajustement du modèle Freely Jointed Chain (FJC) dans chaque région (b). (a) Seules les régions décroissantes sont conservées dans chaque intervalle obtenu en sortie de l'algorithme de lissage polynômial. (b) Les lignes en trait plein correspondent au modèle paramétrique ajusté sur chaque région, et les lignes en pointillé représentent l'extrapolation du modèle (vers le point de contact $z = Z_0$).

1. recherche des points de discontinuité dans la courbe (segmentation d'une courbe de force) et lissage par morceaux de la courbe ;
2. détermination, à partir des résultats du lissage, des régions d'intérêt dans lesquelles les modèles paramétriques s'appliquent ;
3. réalisation finalement de l'ajustement de chaque modèle paramétrique sur la région d'intérêt où il s'applique.

Cette méthodologie est commune au traitement des courbes d'approche et de retrait. Seul le troisième sous-traitement est spécifique aux types d'interactions qui se produisent.

Lissage par morceaux

La première étape de segmentation d'une courbe de force est résumée par la Fig. 1.11. La courbe expérimentale est d'abord prétraitée (soustraction d'une ligne de base affine) de sorte que dans les zones situées à gauche (approche) ou à droite (retrait) de la courbe, la force est nulle. Les résultats de la segmentation (détection des points de discontinuité d'ordres 0, 1 et 2 par sélection de variable vectorielle) sont illustrés dans les deux sous-figures (c) et (f). Dans les deux cas, l'approximation est très précise.

La courbe de retrait présentée est relative à une bactérie en interaction avec une pointe nanométrique. Initialement, le pointe et la bactérie sont en contact (partie gauche de la courbe). Puis, au fur et à mesure que l'on écarte la pointe de la bactérie (z augmente), on observe une poursuite du contact (partie continue à gauche de la courbe), puis des sauts du signal qui indiquent que la pointe perd contact de façon similaire au détachement d'une bande auto-agrippante velcro, avec les protéines adhésives recouvrant le corps de la bactérie, jusqu'à une perte totale du contact (force nulle sur la partie gauche de la courbe). Ce sont ces zones décroissantes qu'il s'agit de segmenter très finement pour pouvoir ensuite ajuster les modèles paramétriques connus. Ici, chaque zone décroissante de la courbe de force est parfaitement segmentée.

Détection des régions d'intérêt

Pour pouvoir ajuster les modèles physiques à une courbe (courbe d'approche ou de retrait), il est essentiel de détecter chaque région d'intérêt dans laquelle les modèles physiques s'appliquent. Pour les courbes d'approche, nous utilisons la règle empirique suivante : le point de contact pointe-échantillon est défini comme le premier point de discontinuité dans la zone où la courbe commence à croître. La zone d'interaction électrostatique (avant contact) est matérialisé sur la Fig. 1.11 (c) par les deux lignes en pointillé, et la zone de contact correspond aux échantillons du signal à droite des lignes en pointillé. Pour les courbes de retrait, les régions d'intérêt sont définies par comme la liste d'intervalles sur lesquels le signal polynômial par morceaux obtenu lors de l'étape de lissage est décroissant.

Ajustement de modèles paramétriques

Une fois les régions d'intérêt détectées, il ne reste plus qu'à ajuster chaque modèle paramétrique dans la région d'intérêt correspondante. Les paramètres physiques sont estimés au sens des moindres carrés, en minimisant un critère du type :

$$\sum_{z \in \mathcal{R}_i} [f(z) - f_i(z; \boldsymbol{\theta}_i)]^2 \tag{1.11}$$

où $\mathcal{R}_i$ désigne la i-ème région d'intérêt détectée et $f_i(z; \boldsymbol{\theta}_i)$ désigne le modèle paramétrique défini sur la i-ème région d'intérêt. $\boldsymbol{\theta}_i$ représente l'ensemble des paramètres de forme qui décrivent la courbe de force (par exemple, la pente pour une partie affine, la constante de temps pour une forme exponentielle, *etc.*). Bien que le nombre de paramètres à estimer soit limité, l'optimisation du critère (1.11) requiert souvent de prendre des précautions, car le critère présente souvent des minima locaux et des zones plates. Nous renvoyons le lecteur au chapitre 4 pour plus de détails.

Les résultats de la détection de régions d'intérêt et de l'ajustement de la courbe de retrait de la Fig. 1.11 (f) sont présentés sur la Fig. 1.12. L'algorithme a été testé sur un très grand nombre de données réelles et fournit des résultats cohérents avec la connaissance de nos collègues physico-chimistes. Exécuté en boucle sur tous les pixels d'une image force-volume, l'algorithme conduit finalement à la reconstruction d'un ensemble d'images 2D représentant chacune l'un des paramètres physiques (topographie, module d'élasticité, *etc.*).

1.4 Organisation des autres chapitres de la thèse

Nous résumons maintenant l'organisation des quatre autres chapitres de cette thèse et leurs contributions principales.

Dans le chapitre 2, nous présentons l'algorithme SBR pour la minimisation de critères ℓ_2-ℓ_0 pénalisés (1.9). C'est un algorithme itératif de type "ajout-retrait" (forward-backward), où à chaque itération, une colonne est soit ajoutée, soit retirée à l'ensemble des colonnes sélectionnées dans le dictionnaire. Ce chapitre permet de faire le lien entre l'algorithme SMLR pour la restauration de processus Bernoulli-Gaussien et les algorithmes de type "ajout-retrait" pour l'approximation parcimonieuse. Nous montrons l'intérêt de l'algorithme SBR sur plusieurs problèmes difficiles dans lesquels les colonnes de la matrice d'observation $\boldsymbol{A}$ sont très corrélées, et notamment sur le problème de la restauration d'un signal polynômial par morceaux. SBR est comparé à des algorithmes gloutons de type ajout (forward), comme l'algorithme Orthogonal Least Squares (OLS). Nous montrons que les performances de SBR sont au moins aussi bonnes, voire meilleures que OLS en un temps de calcul légèrement supérieur.

Dans le chapitre 3, nous présentons l'algorithme de continuation CSBR en tant qu'extension de

SBR. CSBR est défini comme une boucle dans laquelle SBR est appelé récursivement pour des valeurs décroissantes de λ avec pour solution initiale le résultat de SBR obtenu avec la valeur de λ immédiatement supérieure. CSBR fournit donc des approximations de moins en moins parcimonieuses et de plus en plus précises. L'intérêt de CSBR est que les valeurs de λ pour lesquelles SBR est appelé sont calculées de façon adaptative aux données. Moyennant une adaptation mineure, CSBR permet aussi de fournir un ensemble de solutions sous-optimales (pour tout k) pour le problème ℓ_2-ℓ_0 contraint (1.8). Les performances de CSBR sont notamment comparées avec celles de l'algorithme qui LARS basée sur la minimisation d'un critère de type ℓ_2-ℓ_1 (critère (1.8), où la pseudo-norme ℓ_0 est remplacée par la norme ℓ_1).

Dans le chapitre 4, nous appliquons l'algorithme CSBR à l'approximation d'une courbe de force en microscopie de force atomique. L'algorithme consiste dans un premier temps à lisser la courbe de force par un signal polynômial par morceaux, de façon à segmenter les régions d'intérêt sur la courbe de force, puis dans un second temps, à estimer les paramètres physiques (élasticité, topographie, *etc.*) par une procédure de régression appliquée sur chaque région d'intérêt. Ce chapitre est un projet d'article dans une revue appliquée. Par conséquent, pour des raisons pédagogiques, nous nous sommes attaché à décrire les algorithmes dans les grandes lignes.

Enfin, le chapitre 5 est consacré au traitement conjoint d'un ensemble de courbes de force, par opposition au traitement indépendant du chapitre 4. Ce traitement conjoint repose sur une procédure de séparation de sources retardées, où chaque source est décrite par un signal polynômial par morceaux. Bien que la motivation de l'algorithme soit d'ordre applicatif, nous nous efforçons de positionner l'algorithme par rapport à la littérature de la séparation de sources parcimonieuse et de la séparation de sources retardées (appelée *anechoic source separation* en traitement de signal acoustique). Nous discutons les conditions d'identifiabilité du problème de séparation à partir d'un grand nombre de mélanges, puis nous présentons un algorithme basé sur la mise en correspondance des descriptions parcimonieuses de l'ensemble des mélanges. Cet algorithme est d'abord décrit dans un cadre sans bruit, puis étendu au cadre bruité.

Chapitre 2

From Bernoulli-Gaussian deconvolution to sparse signal restoration

Charles Soussen* [1], Jérôme Idier [2], *Member, IEEE*, David Brie[1] and Junbo Duan[1]

Abstract

Formulated as a least-square problem under an ℓ_0 constraint, sparse signal restoration is a discrete optimization problem, known to be NP complete. Classical algorithms include, by increasing cost and efficiency, Matching Pursuit (MP), Orthogonal Matching Pursuit (OMP), Orthogonal Least Squares (OLS) and the exhaustive search algorithm. In inverse problems involving highly correlated dictionaries, OMP and OLS are not guaranteed to find the optimal solution. It is of interest to develop slightly slower sub-optimal search algorithms yielding better approximations. We revisit the Single Most Likely Replacement (SMLR) algorithm, developed in the mid-80's for Bernoulli-Gaussian signal restoration. We show that the formulation of sparse signal restoration as a limit case of Bernoulli-Gaussian signal restoration leads to an ℓ_0-penalized least-square minimization problem, to which SMLR can be straightforwardly adapted. The resulting algorithm, called Single Best Replacement (SBR), can be interpreted as a forward-backward extension of OLS. A fast and stable implementation is proposed. The approach is illustrated on a deconvolution problem with a Gaussian impulse response and on the joint detection of discontinuities at different orders in a signal.

Keywords

Sparse signal estimation ; inverse problems ; Bernoulli-Gaussian signal restoration ; SMLR algorithm ; mixed ℓ_2-ℓ_0 criterion minimization ; Orthogonal Least Squares ; forward-backward greedy algorithms.

1. C. Soussen, D. Brie, and J. Duan are with the Centre de Recherche en Automatique de Nancy (CRAN, UMR 7039, Nancy-University, CNRS). Boulevard des Aiguillettes, B.P. 70239, F-54506 Vandoeuvre-lès-Nancy, France. Tel : (+33)-3 83 68 44 71, Fax : (+33)-3 83 68 44 62. E-mail : FirstName.SecondName@cran.uhp-nancy.fr.

2. J. Idier is with the Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN, UMR CNRS 6597), BP 92101, 1 rue de la Noë, 44321 Nantes Cedex 3, France. Tel : (+33)-2 40 37 69 09, Fax : (+33)-2 40 37 69 30. E-mail : Jerome.Idier@irccyn.ec-nantes.fr.

2.1 Introduction

Sparse signal restoration arises in inverse problems such as Fourier synthesis, mono- and multidimensional deconvolution, and statistical regression. It consists in the decomposition of a given signal $\boldsymbol{y}$ as a linear combination of a limited number of elements from a dictionary $\boldsymbol{A}$. While formally very similar, sparse signal restoration has to be distinguished from sparse signal approximation. The main difference is that in sparse signal restoration, the choice of the dictionary is imposed by the inverse problem at hand whereas in sparse signal approximation, the dictionary has to be chosen according to its ability to represent the data with a limited number of coefficients. A more subtle difference is that in sparse signal restoration, the focus is set on the estimation of the weights of the linear combination while in sparse signal approximation, the goal is to reproduce the data $\boldsymbol{y}$ as well as possible at a given level of sparsity.

Sparse signal restoration can be formulated as the minimization of a least-square cost function of the form $\mathcal{E}(x) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ under the constraint that the ℓ_0 pseudo-norm of $\boldsymbol{x}$, defined as the number of non-zero entries in $\boldsymbol{x}$, is lower than a given number k. This problem is often referred to as *subset selection*, because imposing the sparsity constraint consists in selecting a subset of columns of $\boldsymbol{A}$. This yields a discrete problem (since there are a finite number of possible subsets) which is known to be NP-complete [12]. In this paper, we focus on "difficult" problems in which some of the columns of $\boldsymbol{A}$ are highly correlated, the unknown weight vector $\boldsymbol{x}^*$. is only approximately sparse, and/or the data are noisy. Hereafter, we distinguish two approaches to address the subset selection problem in a fast and sub-optimal manner and we discuss their relevance for difficult problems.

The first approach, which has been the most popular in the last decade, approximates the subset selection problem by a continuous optimization problem, convex or not, that is easier to solve [27], [28], [29], [30]. In particular, the approach that replaces the ℓ_0-norm by the ℓ_1-norm [30], [29] has been increasingly investigated, leading to the LASSO optimization problem. Its popularity relies on efficient algorithms, such as LARS which finds the set of solutions for all degrees of sparsity [8]. Several authors have provided sufficient conditions under which the ℓ_0- and ℓ_1-constrained leastsquare problems lead to solutions having the same support [30], [31]. These conditions typically state that the unknown weight signal has to be highly sparse, that the correlation between any pair of columns of $\boldsymbol{A}$ must be sufficiently small, and that the noise level must be low. They are often not satisfied when dealing with real data.

The second approach addresses the *exact* subset selection problem using either thresholding algorithms [32], [33] or greedy search algorithms. The latter gradually increase or decrease by one the set of active columns. The simplest greedy algorithms are Matching Pursuit (MP) [24] and the improved version Orthogonal Matching Pursuit (OMP) [25]. Both are referred to as forward greedy algorithms, since they start from an empty active set and then gradually increase it by one element. In contrast, the backward algorithm of Couvreur and Bresler [34] starts from a complete active set which is gradually decreased by one element. It is, however, only valid if the dictionary is not overcomplete. A few authors have introduced forward-backward algorithms in which insertions and removals of dictionary elements into/from the active set are both allowed [35], [36]. This strategy yields better recovery performance since an early wrong selection can be counteracted by its further removal from the active set. In contrast, the insertion of a wrong element is irreversible when using forward algorithms.

The choice of the algorithm depends on the amount of time available and on the structure of matrix $\boldsymbol{A}$. In favorable cases, the sub-optimal search algorithms described above (belonging to the first or the second approach) provide solutions having the same support as the exhaustive search solution. For example, if the unknown signal $\boldsymbol{x}^*$ is highly sparse and if the correlation between any pair of columns of $\boldsymbol{A}$ is low, the ℓ_1-norm approximation provides optimal solutions [30], [31]. In other cases, however, the only guarantee to recover the optimal solution is to use the exhaustive search algorithm. When fast sub-optimal algorithms lead to unsatisfactory results, it is of great

interest to develop slower sub-optimal algorithms providing more accurate solutions, but remaining very fast compared to the exhaustive search. The Orthogonal Least Squares algorithm (OLS) [23] which is sometimes confused with OMP [37], falls into this category. Both OLS and OMP share the same structure, the difference being that at each iteration, OLS solves a large number of least-square problems ($n-k$, where k is the cardinal of the current active set) while OMP only computes the $n-k$ inner products between the current residual $\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}$ and the candidate columns $\boldsymbol{a}_i$ and chooses the column yielding the maximal inner product. OMP solves only one least-square problem per iteration, once the column to be inserted is selected (in order to update all the active weights). In the following, we propose a forward-backward extension of OLS allowing an insertion or a removal at each iteration, each iteration requiring to solve n least-square problems. It differs from the FoBa algorithm of Zhang [36] which is an OMP forward-backward extension. It is closer to the bidirectional OLS based algorithm of Haugland [35], the main differences relying on the search and implementation strategies.

The starting point of our forward-backward algorithm is the Single Most Likely Replacement (SMLR) algorithm, which proved to be a very efficient tool for the deconvolution of a sparse signal modeled as a Bernoulli-Gaussian process [38], [39], [22], [13]. This approach relies on a Bayesian formulation of a deconvolution problem of the form $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{n}$ (where $\boldsymbol{A}$ denotes the convolution matrix) and on the maximum *a posteriori* (MAP) estimation of the sparse signal. The Bernoulli-Gaussian model is a probabilistic model for sparse signals, in which (binary) Bernoulli random variables are associated to the position of the non zero entries in $\boldsymbol{x}$ while the corresponding amplitudes are distributed according to an independent identically distributed (i.i.d.) centered Gaussian distribution of variance σ_x^2. SMLR is a deterministic ascent algorithm which maximizes the posterior likelihood in a sub-optimal manner. It consists in updates (increase or decrease) of the support of $\boldsymbol{x}$ by one element and the subsequent estimation of the non-zero amplitudes. Sparse signal restoration can be seen as a limit case of Bernoulli-Gaussian MAP restoration in which the variance σ_x^2 of the amplitudes is set to infinity. We will deduce an adaptation of SMLR to subset selection relying on a single insertion or a single removal of a column into/from the active set.

The paper is organized as follows. In Section 2.2, we introduce the Bernoulli-Gaussian model and the Bayesian framework from which we formulate the sparse signal restoration problem. In Section 2.3, we adapt the SMLR algorithm resulting in the so-called Single Best Replacement (SBR) algorithm. In Section 2.4, a fast and stable SBR implementation is proposed, based on the efficient update of the squared error when the active set is modified by one element. Finally, Sections 2.5 and 2.6 illustrate the method on the sparse spike deconvolution with a Gaussian impulse response and on the joint detection of discontinuities at different orders in a signal.

2.2 From Bernoulli-Gaussian signal modeling to sparse signal representation

We consider the restoration of a sparse signal $\boldsymbol{x}$ from a linear observation $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{n}$, where $\boldsymbol{n}$ stands for the observation noise. An acknowledged probabilistic model dedicated to sparse signals is the Bernoulli-Gaussian (BG) model [38], [39], [13]. For such model, deterministic optimization algorithms [13] and Markov chain Monte Carlo techniques [40] are used to compute the MAP and the posterior mean, respectively. We first recall the known BG models and the formulation of BG signal restoration in the Bayesian framework. Then, we extend this formulation to a more general representation of sparse signals.

2.2.1 Preliminary definitions and working assumptions

Given an observation vector $\boldsymbol{y} \in \mathbb{R}^m$ and a dictionary $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n] \in \mathbb{R}^{m \times n}$, a subset selection algorithm aims at computing a weight vector $\boldsymbol{x} \in \mathbb{R}^n$ yielding an accurate approximation $\boldsymbol{y} \approx \boldsymbol{A}\boldsymbol{x}$ of the observation. The columns $\boldsymbol{a}_i$ of $\boldsymbol{A}$ whose indices correspond to the non-zero components x_i of $\boldsymbol{x}$ are referred to as the active (or selected) columns.

Throughout this paper, no assumption is made on the size of $\boldsymbol{A}$: m can be either smaller or larger than n. Here, $\boldsymbol{A}$ is assumed to satisfy the unique representation property (URP). This assumption is usual in the case $m \leqslant n$ [41]. It is a stronger assumption than the full rank assumption. We now recall this definition and extend it to the case where $m > n$. Notation $\|\cdot\|$ refers to the Euclidean norm.

Definition 1. *When $m \leqslant n$, $\boldsymbol{A}$ satisfies the URP* if and only if *any selection of m columns of $\boldsymbol{A}$ forms a family of linearly independent vectors. When $m > n$, $\boldsymbol{A}$ satisfies the URP* if and only if *it is full rank.*

Before going further, let us mention that this assumption can be relaxed providing that the search strategy can guarantee that the selected columns of $\boldsymbol{A}$ result in a full rank matrix (see Section 2.6.3 for details).

Under the URP assumption, when $m \leqslant n$, the system $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ has a number of solutions whose ℓ_0-norm are lower or equal to m since any selection of m columns yields a solution of the system. When $m > n$, there is generally no solution to $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ but the least-square estimator $\boldsymbol{x} = (\boldsymbol{A}^t\boldsymbol{A})^{-1}\boldsymbol{A}^t\boldsymbol{y}$ is unique, although not necessarily sparse.

Definition 2. *The support of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is the set $\mathcal{S}(\boldsymbol{x}) \subseteq \{1, \ldots, n\}$ defined by $i \in \mathcal{S}(\boldsymbol{x})$* if and only if $x_i \neq 0$.

Definition 3. *We denote by $\mathcal{Q} \subseteq \{1, \ldots, n\}$ the active set and we define the related vector $\boldsymbol{q} \in \{0,1\}^n$ by $q_i = 1$* if and only if $i \in \mathcal{Q}$. *Let $\boldsymbol{A}_\mathcal{Q}$ be the submatrix of size $m \times \mathrm{Card}\,[\mathcal{Q}]$ formed of the active columns of $\boldsymbol{A}$ ($\boldsymbol{a}_i, i \in \mathcal{Q}$). The observation model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$ also reads $\boldsymbol{y} = \boldsymbol{A}_\mathcal{Q}\boldsymbol{t} + \boldsymbol{n}$, where the reduced vector $\boldsymbol{t}$ of size $\mathrm{Card}\,[\mathcal{Q}]$ gathers the values $\{x_i, i \in \mathcal{Q}\}$.*

Definition 4. *For all $\mathcal{Q} \subseteq \{1, \ldots, n\}$ such that $\mathrm{Card}\,[\mathcal{Q}] \leqslant \min(m, n)$, let $\boldsymbol{x}_\mathcal{Q}$ be the least-square solution and let $\mathcal{E}_\mathcal{Q}$ be the associated squared error :*

$$\boldsymbol{x}_\mathcal{Q} \triangleq \underset{\mathcal{S}(\boldsymbol{x}) \subseteq \mathcal{Q}}{\arg\min} \{\mathcal{E}(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2\} \tag{2.1}$$

$$\mathcal{E}_\mathcal{Q} \triangleq \mathcal{E}(\boldsymbol{x}_\mathcal{Q}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\mathcal{Q}\|^2 \tag{2.2}$$

2.2.2 Bernoulli-Gaussian models

A BG process [1] $\boldsymbol{x}$ can be defined by means of a Bernoulli random vector $\boldsymbol{q} \in \{0,1\}^n$ corresponding to the active set, and a Gaussian random vector $\boldsymbol{r} \sim \mathcal{N}(\boldsymbol{0}, \sigma_x^2 \boldsymbol{I}_n)$, where $\boldsymbol{I}_n$ stands for the identity matrix of size $n \times n$. Each sample x_i of $\boldsymbol{x}$ is modeled as $x_i = q_i r_i$ [38], [39]. Thus, r_i code for the amplitudes of the nonzero entries in $\boldsymbol{x}$. In the vector form, $\boldsymbol{x}$ reads $\boldsymbol{x} = \boldsymbol{\Delta}_\boldsymbol{q}\boldsymbol{r}$ where $\boldsymbol{\Delta}_\boldsymbol{q}$ is the diagonal matrix of size $n \times n$ whose diagonal elements are equal to q_i. The Bernoulli random variables q_i code for the presence ($q_i = 1$) or absence ($q_i = 0$) of signal at location i, the Bernoulli parameter $\rho = \Pr(q_i = 1)$ being the probability of presence of signal. It is easy to check that the prior likelihood of $\boldsymbol{q}$ reads $l(\boldsymbol{q}) = \rho^{\|\boldsymbol{q}\|_0}(1-\rho)^{n-\|\boldsymbol{q}\|_0}$. Because $\boldsymbol{q}$ and $\boldsymbol{r}$ are independent random vectors, the prior likelihood of $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{r})$ reads :

$$l(\boldsymbol{q}, \boldsymbol{r}) = l(\boldsymbol{r})l(\boldsymbol{q}) = g(\boldsymbol{r}; \sigma_x^2 \boldsymbol{I}_n)\rho^{\|\boldsymbol{q}\|_0}(1-\rho)^{n-\|\boldsymbol{q}\|_0} \tag{2.3}$$

1. For convenience, we use the same notations for random vectors and their realization.

where $g(.;\mathbf{\Gamma})$ denotes the probability density of the centered Gaussian with covariance matrix $\mathbf{\Gamma}$.

2.2.3 Bayesian formulation of sparse signal restoration

The Bayesian formulation of the inverse problem $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}+\boldsymbol{n}$ consists in inferring the distribution of $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{r})$ knowing $\boldsymbol{y}$. The MAP estimator of $\boldsymbol{x}$ can be obtained by maximizing the marginal distribution $l(\boldsymbol{q}|\boldsymbol{y})$ [13] or the joint distribution $l(\boldsymbol{q}, \boldsymbol{r}|\boldsymbol{y})$ [39], [22]. Following [39], we focus on the joint likelihood $l(\boldsymbol{q}, \boldsymbol{r}|\boldsymbol{y})$ which leads to a cost function involving the squared error $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ and the ℓ_0-norm of $\boldsymbol{x}$.

Assuming a Gaussian white noise $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \sigma_n^2 \boldsymbol{I}_m)$, independent from $\boldsymbol{x}$, it is easy to obtain

$$\begin{aligned} \mathcal{L}(\boldsymbol{q}, \boldsymbol{r}) &\triangleq -2\sigma_n^2 \log[l(\boldsymbol{q}, \boldsymbol{r}|\boldsymbol{y})] \\ &= \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\Delta}_{\boldsymbol{q}}\boldsymbol{r}\|^2 + \frac{\sigma_n^2}{\sigma_x^2}\|\boldsymbol{r}\|^2 + \lambda\|\boldsymbol{q}\|_0 + C \end{aligned} \tag{2.4}$$

where $\lambda = 2\sigma_n^2 \log(1/\rho - 1)$ and C is a constant. Given $\boldsymbol{q}$, let us split $\boldsymbol{r}$ into two subvectors $\boldsymbol{u}$ and $\boldsymbol{t}$ indexed by the null and non-null entries of $\boldsymbol{q}$, respectively. Since $\boldsymbol{A}\Delta_{\boldsymbol{q}}\boldsymbol{r} = \boldsymbol{A}_{\mathcal{Q}}\boldsymbol{t}$ does not depend on $\boldsymbol{u}$, it is easy to check that $\min_{\boldsymbol{u}} \mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{u}) = \mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0})$. Finally, the joint MAP estimation problem reduces to the minimization of $\mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0})$ w.r.t. ($\boldsymbol{q}$,$\boldsymbol{t}$).

2.2.4 Mixed ℓ_2-ℓ_0 minimization as a limit case

A signal $\boldsymbol{x}$ is sparse if some entries x_i are equal to 0. Since this definition does not impose constraints on the range of values of the non zero amplitudes, we choose to describe a sparse signal by a limit Bernoulli-Gaussian model in which the amplitude variance σ_x^2 is set to infinity. The minimization of $\mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0})$ thus rereads :

$$\min_{\boldsymbol{q},\boldsymbol{t}} \{\mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0}) = \|\boldsymbol{y} - \boldsymbol{A}_{\mathcal{Q}}\boldsymbol{t}\|^2 + \lambda\|\boldsymbol{q}\|_0\} \tag{2.5}$$

Theorem 1. *The above formulation (2.5) is equivalent to :*

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \{\mathcal{J}(\boldsymbol{x}; \lambda) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{x}\|_0\} \tag{2.6}$$

which is referred to as the ℓ_0-penalized least-square problem. The term "equivalent" means that given a minimizer ($\boldsymbol{q}$,$\boldsymbol{t}$) of (2.5), the related vector $\boldsymbol{x} = \{\boldsymbol{t}, \boldsymbol{0}\}$ is a minimizer of (2.6), and conversely, given a minimizer $\boldsymbol{x}$ of (2.6), the vectors $\boldsymbol{q}$ and $\boldsymbol{t}$ defined as the support of $\boldsymbol{x}$ and its non-zero amplitudes, respectively, are such that ($\boldsymbol{q}$,$\boldsymbol{t}$) is a minimizer of (2.5).

Proof. To prove the equivalence, we first prove that $\min_{\boldsymbol{x}} \mathcal{J}(\boldsymbol{x}; \lambda) = \min_{\boldsymbol{q},\boldsymbol{t}} \mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0})$:

- Let $\boldsymbol{x}$ be a minimizer of $\mathcal{J}(.; \lambda)$. We set $\boldsymbol{q}$ and $\boldsymbol{t}$ to the support and the non zero amplitudes of $\boldsymbol{x}$, respectively. Obviously, $\mathcal{J}(\boldsymbol{x}; \lambda) = \mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0})$. It follows that $\min_{\boldsymbol{x}} \mathcal{J}(\boldsymbol{x}; \lambda) \geqslant \min_{\boldsymbol{q},\boldsymbol{t}} \mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0})$.
- Let ($\boldsymbol{q}$,$\boldsymbol{t}$) be a minimizer of $\mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0})$. The vector $\boldsymbol{x} = \{\boldsymbol{t}, \boldsymbol{0}\}$ is such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}_{\boldsymbol{Q}}\boldsymbol{t}$ and $\|\boldsymbol{x}\|_0 = \|\boldsymbol{t}\|_0 \leqslant \|\boldsymbol{q}\|_0$. Therefore, $\mathcal{J}(\boldsymbol{x}; \lambda) \leqslant \mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0})$. It follows that $\min_{\boldsymbol{q},\boldsymbol{t}} \mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0}) \geqslant \min_{\boldsymbol{x}} \mathcal{J}(\boldsymbol{x}; \lambda)$.

We have shown that $\min_{\boldsymbol{x}} \mathcal{J}(\boldsymbol{x}; \lambda) = \min_{\boldsymbol{q},\boldsymbol{t}} \mathcal{L}(\boldsymbol{q}, \boldsymbol{t}, \boldsymbol{0})$, but also that the minimizers of both problems coincide, *i.e.,* are vectors describing identical signals. □

In the following, we focus on the minimization problem (2.6) involving the penalization term $\|\boldsymbol{x}\|_0$. The hyperparameter λ is fixed. It controls the level of sparsity of the desired solution. The

algorithm that will be developed is based on an efficient search of the support of $\boldsymbol{x}$. In that respect, the ℓ_0-penalized least-square problem does not drastically differ from the ℓ_0-constrained problem $\min \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ subject to $\|\boldsymbol{x}\|_0 \leqslant k$.

2.3 Adaptation of SMLR to ℓ_0-penalized least-square optimization

We propose to adapt the SMLR algorithm to the minimization of the mixed ℓ_2-ℓ_0 cost function $\mathcal{J}(\boldsymbol{x};\lambda)$ defined in (2.6). To clearly distinguish SMLR which specifically aims at minimizing (2.4), the adapted algorithm will be termed as Single Best Replacement (SBR).

2.3.1 Principle of SMLR and main notations

The Single Most Likely Replacement algorithm [38] is a deterministic coordinatewise ascent algorithm to maximize log-likelihood functions of the form $l(\boldsymbol{q}|\boldsymbol{y})$ (marginal MAP estimation) or $l(\boldsymbol{q},\boldsymbol{t},\boldsymbol{0}|\boldsymbol{y})$ (joint MAP estimation). In the latter case, it is easy to check from (2.4) that given $\boldsymbol{q}$, the maximizer of $l(\boldsymbol{q},\boldsymbol{t},\boldsymbol{0}|\boldsymbol{y})$ w.r.t. $\boldsymbol{t}$ has a closed form expression $\boldsymbol{t} = \boldsymbol{t}(\boldsymbol{q})$. Consequently, the joint MAP estimation reduces to the maximization of $l(\boldsymbol{q},\boldsymbol{t}(\boldsymbol{q}),\boldsymbol{0}|\boldsymbol{y})$ w.r.t. $\boldsymbol{q}$. At each SMLR iteration, all the possible single replacements of the support $\boldsymbol{q}$ (set $q_i = 1 - q_i$ while keeping the other q_j, $j \neq i$ unchanged) are tested, then the replacement yielding the maximal increase of $l(\boldsymbol{q},\boldsymbol{t}(\boldsymbol{q}),\boldsymbol{0}|\boldsymbol{y})$ is chosen. This task is repeated until no single replacement can increase $l(\boldsymbol{q},\boldsymbol{t}(\boldsymbol{q}),\boldsymbol{0}|\boldsymbol{y})$ anymore.

The number of possible supports $\boldsymbol{q}$ being finite (2^n) and SMLR being an ascent algorithm, it terminates after a finite number of iterations.

Before adapting SMLR, let us introduce some useful notations.

– We denote by $\boldsymbol{Q} \bullet i$ a single replacement, *i.e.,* the insertion or removal of an index i into/from the active set $\mathcal{Q}$:

$$\mathcal{Q} \bullet i \triangleq \begin{cases} \mathcal{Q} \cup \{i\}, & \text{if } i \notin \mathcal{Q}\,; \\ \mathcal{Q} \backslash \{i\}, & \text{otherwise.} \end{cases} \tag{2.7}$$

– If $\text{Card}\,[\mathcal{Q}] \leqslant \min(m,n)$, we define the cost functions :

$$\mathcal{J}_{\mathcal{Q}}(\lambda) \triangleq \mathcal{J}(\boldsymbol{x}_{\mathcal{Q}};\lambda) = \mathcal{E}_{\mathcal{Q}} + \lambda\|\boldsymbol{x}_{\mathcal{Q}}\|_0 \tag{2.8}$$

$$\mathcal{K}_{\mathcal{Q}}(\lambda) = \mathcal{E}_{\mathcal{Q}} + \lambda\text{Card}\,[\mathcal{Q}] \tag{2.9}$$

where the least-square solution $\boldsymbol{x}_{\mathcal{Q}}$ and the corresponding error $\mathcal{E}_{\mathcal{Q}}$ have been defined in (2.1) and (2.2). Obviously, $\mathcal{J}_{\mathcal{Q}}(\lambda) = \mathcal{K}_{\mathcal{Q}}(\lambda)$ *if and only if* $\boldsymbol{x}_{\mathcal{Q}}$ has a support equal to $\mathcal{Q}$. In subsection 2.3.2, we introduce a first version of SBR involving $\mathcal{J}_{\mathcal{Q}}(\lambda)$ only, and in subsection 2.3.3, we present an alternative (simpler) version relying on the computation of $\mathcal{K}_{\mathcal{Q}}(\lambda)$ instead of $\mathcal{J}_{\mathcal{Q}}(\lambda)$ and we discuss in which extent both versions differ. Then, subsection 2.3.4 describes the behavior of SBR and states its main properties.

2.3.2 The Single Best Replacement algorithm (first version)

SMLR can be seen as an exploration strategy for discrete optimization rather than an algorithm specific to a posterior likelihood function. Here, we use the same strategy to minimize the cost function $\mathcal{J}(\boldsymbol{x};\lambda)$. We rename the algorithm Single Best Replacement to remove any statistical connotation. The SBR algorithm works as follows. At each iteration, the n possible single replacements $\mathcal{Q} \bullet i (i = 1,\ldots,n)$ are tested, then the best is selected, *i.e.,* the replacement yielding the maximal decrease of $\mathcal{J}(\boldsymbol{x};\lambda)$. This task is repeated until $\mathcal{J}(\boldsymbol{x};\lambda)$ cannot decrease anymore. Let us detail an SBR iteration.

TABLE 2.1 – SBR algorithm (final version). By default, the initial active set is empty : $\mathcal{Q}_1 = \emptyset$

Input : $\boldsymbol{A}$, $\boldsymbol{y}$, λ and active set $\mathcal{Q}_1(\text{Card}\,[\mathcal{Q}]_1 \leqslant \min(m,n))$
Step 1 : Set $j = 1$.
Step 2 : For $i \in \{1,\ldots,n\}$, compute $\mathcal{K}_{\mathcal{Q}_j \bullet i}(\lambda)$.
Compute ℓ using (2.11).
If $\mathcal{K}_{\mathcal{Q}_j \bullet \ell} < \mathcal{K}_{\mathcal{Q}_j}(\lambda)$,
Set $\mathcal{Q}_{j+1} = \mathcal{Q}_j \bullet \ell$
else,
Terminate SBR.
End if.
Set $j = j + 1$ and go to Step 2.
Output : active set $\mathcal{Q}_j$ =SBR$(\mathcal{Q}_1; \lambda)$

Consider the current active set $\mathcal{Q}$. For each index i, we compute the minimizer $\boldsymbol{x}_{\mathcal{Q}\bullet i}$ of $\mathcal{E}$ whose support is included in $\mathcal{Q} \bullet i$ and we keep in memory the value of $\mathcal{J}_{\mathcal{Q}\bullet i}(\lambda)$. If the minimum of $\{\mathcal{J}_{\mathcal{Q}\bullet i}(\lambda), i = 1,\ldots,n\}$ is lower than $\mathcal{J}_{\mathcal{Q}}(\lambda)$, then we select the index yielding this minimal value :

$$\ell \in \underset{i\in\{1,\ldots,n\}}{\arg\min}\, \mathcal{J}_{\mathcal{Q}\bullet i}(\lambda) \tag{2.10}$$

The next SBR iterate is thus defined as $\mathcal{Q}' = \mathcal{Q} \bullet \ell$, yielding the vector $\boldsymbol{x}_{\mathcal{Q}'}$.

Except when an initial support estimate (of cardinality lower than $\min(m,n)$) is available, we suggest to use an initial empty active set.

Remark 1 (**Relationship between SBR and SMLR**). *We introduced SBR as the application of the SMLR search strategy to the ℓ_0-penalized least square cost function (2.6) which is obtained by taking the limit of the cost function (2.4) when σ_x tends towards infinity. Conversely, applying SMLR to (2.4) and then, taking the limit of the SMLR formula when σ_x tends to infinity also yields the SBR algorithm.*

Actually, the main difference between SMLR and SBR is that SMLR (which can take several forms depending on the use of the joint distribution $l(\boldsymbol{q},\boldsymbol{r}|\boldsymbol{y})$ or the marginal distribution $l(\boldsymbol{q}|\boldsymbol{y})$) involves the inversion of a matrix of the form $\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{A}_{\mathcal{Q}} + \alpha\boldsymbol{I}_{\text{Card}[\mathcal{Q}]}$ whereas SBR involves the inverse of the Gram matrix $\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{A}_{\mathcal{Q}}$. For this reason, instabilities may occur while using SBR when $\boldsymbol{A}_{\mathcal{Q}}$ is ill conditioned. The use of the term $\alpha\boldsymbol{I}_{\text{Card}[\mathcal{Q}]}$, which acts as a regularization on the amplitude values, avoids such instability while using SMLR at the price of handling the additional hyperparameter α.

2.3.3 Modified version of SBR (final version)

We introduce a slight modification of SBR by replacing (2.10) with :

$$\ell \in \underset{i\in\{1,\ldots,n\}}{\arg\min}\, \mathcal{K}_{\mathcal{Q}\bullet i}(\lambda) \tag{2.11}$$

We propose this modification because $\mathcal{K}_{\mathcal{Q}}(\lambda) = \mathcal{E}_{\mathcal{Q}} + \lambda\text{Card}\,[\mathcal{Q}]$ can be computed more efficiently than $\mathcal{J}_{\mathcal{Q}}(\lambda)$, the computation of $\boldsymbol{x}_{\mathcal{Q}}$ being no longer necessary. The use of $\mathcal{K}_{\mathcal{Q}}(\lambda)$ makes the penalization term very easy to update when $\mathcal{Q}$ is modified by one element (add or subtract λ), and the only necessary update is that of $\mathcal{E}_{\mathcal{Q}}$. We now show that there is almost surely no difference between both versions of SBR provided that the data $\boldsymbol{y}$ are corrupted with "non degenerate" noise.

Theorem 2. *Let $\boldsymbol{y} = \boldsymbol{y}_0 + \boldsymbol{n}$, where $\boldsymbol{y}_0$ is a given vector of $\mathbb{R}^m$ and $\boldsymbol{n}$ is a random vector. We*

assume that $\boldsymbol{n}$ *is an absolute continuous random vector, i.e., admitting a probability density w.r.t. the Lebesgue measure. Then, when* $\mathrm{Card}\,[\mathcal{Q}] \leqslant \min(m,n)$*, the probability that* $\|\boldsymbol{x}_{\mathcal{Q}}\|_0 < \mathrm{Card}\,[\mathcal{Q}]$ *is equal to 0, i.e.,* $\|\boldsymbol{x}_{\mathcal{Q}}\|_0 = \mathrm{Card}\,[\mathcal{Q}]$ *almost surely.*

Proof. Let $k = \mathrm{Card}\,[\mathcal{Q}]$ and let $\boldsymbol{t}_{\mathcal{Q}}$ be the minimizer of $\|\boldsymbol{y} - \boldsymbol{A}_{\mathcal{Q}}\boldsymbol{t}\|^2$ over $\mathbb{R}^k$. Obviously, $\|\boldsymbol{x}_{\mathcal{Q}}\|_0 = \|\boldsymbol{t}_{\mathcal{Q}}\|_0 \leqslant k$. Let $\boldsymbol{V}_{\mathcal{Q}} = (\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{A}_{\mathcal{Q}})^{-1}\boldsymbol{A}_{\mathcal{Q}}^t$ be the matrix of size $k \times m$ such that $\boldsymbol{t}_{\mathcal{Q}} = \boldsymbol{V}_{\mathcal{Q}}\boldsymbol{y}$. Denoting by $\boldsymbol{v}^1, \ldots, \boldsymbol{v}^k \in \mathbb{R}^m$ the row vectors of $\boldsymbol{V}_{\mathcal{Q}}$, $\|\boldsymbol{t}_{\mathcal{Q}}\|_0 < k$ *if and only if* there exists i such that $\langle \boldsymbol{y}, \boldsymbol{v}^i \rangle = 0$ (where $\langle ., . \rangle$ denotes the inner product). Because $\boldsymbol{A}_{\mathcal{Q}}$ is full rank, $\boldsymbol{V}_{\mathcal{Q}}$ is full rank and then $\forall i, \boldsymbol{v}^i \neq \boldsymbol{0}$. Denoting by $\mathcal{H}^{\perp}(\boldsymbol{v}^i)$ the hyperplane of $\mathbb{R}^m$ which is orthogonal to $\boldsymbol{v}^i$, we have

$$\|\boldsymbol{x}_{\mathcal{Q}}\|_0 < k \Leftrightarrow \boldsymbol{y} \in \bigcup_{i=1}^{k} \mathcal{H}^{\perp}(\boldsymbol{v}^i) \tag{2.12}$$

Because the set $\bigcup_i \mathcal{H}^{\perp}(\boldsymbol{v}^i)$ has a Lebesgue measure equal to zero and the random vector $\boldsymbol{y}$ admits a probability density, the probability of event (2.12) is zero, thus $\Pr(\|\boldsymbol{x}_{\mathcal{Q}}\|_0 < k) = 0$. □

Theorem 2 implies that when dealing with real noisy data, it is almost sure that no active component x_i is exactly equal to 0. Thus, the original and modified versions of SBR almost surely lead to exactly the same iterates. Even in the noiseless case, an active component is rarely numerically evaluated to 0 due to the round-off errors. In all cases, the modified version of SBR can be applied without restriction and the properties stated below (*e.g.,* termination after a finite number of iterations) remain valid even if an SBR iterate satisfies $\|\boldsymbol{x}_{\mathcal{Q}}\|_0 < \mathrm{Card}\,[\mathcal{Q}]$.

We adopt the modified version of SBR in the rest of the paper. It is summarized in Tab. 2.1.

2.3.4 Behavior and adaptations of SBR

Termination : SBR is a descent algorithm because the value of $\mathcal{K}_{\mathcal{Q}}(\lambda)$ is always decreasing. Consequently, a set $\mathcal{Q}$ cannot be explored twice and similarly to SMLR, SBR terminates after a finite number of iterations. Notice that the size of $\mathcal{Q}$ remains lower or equal to $\min(m,n)$. Indeed, if a set $\mathcal{Q}$ of cardinality $\min(m,n)$ is reached, then $\mathcal{E}_{\mathcal{Q}}$ is equal to 0 due to the URP assumption. Hence, any replacement of the form $\mathcal{Q}' = \mathcal{Q} \cup \{i\}$ yields an increase of the cost function ($\mathcal{K}_{\mathcal{Q}'}(\lambda) = \mathcal{K}_{\mathcal{Q}}(\lambda) + \lambda$). We emphasize that no stopping condition is needed unlike many algorithms which require to set a maximum number of iterations (MP and variations, OLS) and/or a threshold on the squared error variation (CoSaMP, Iterative Hard Thresholding).

Proposition 1. *Under the assumptions of Theorem 2, each SBR iterate* $\boldsymbol{x}_{\mathcal{Q}}$ *is almost surely a local minimizer of* $\mathcal{J}(\boldsymbol{x};\lambda)$ *and of the* ℓ_0*-constrained problem* $\min \mathcal{E}(\boldsymbol{x})$ *subject to* $\|\boldsymbol{x}\|_0 \leqslant k$*, with* $k = \mathrm{Card}\,[\mathcal{Q}]$*. This property holds in particular for the SBR output.*

Proof. Let $\boldsymbol{x} = \boldsymbol{x}_{\mathcal{Q}}$ be an SBR iterate. According to Theorem 2, the support $\mathcal{S}(\boldsymbol{x}) = \mathcal{Q}$ almost surely. Setting $\varepsilon = \min_{i \in \mathcal{Q}} |x_i| > 0$, it is easy to check that if $\boldsymbol{x}' \in \mathbb{R}^n$ satisfies $\|\boldsymbol{x}' - \boldsymbol{x}\| < \varepsilon$, then $\forall i \in \mathcal{Q}, x'_i \neq 0$. Thus, $\|\boldsymbol{x}' - \boldsymbol{x}\| < \varepsilon$ implies that $\mathcal{S}(\boldsymbol{x}') \supseteq \mathcal{S}(\boldsymbol{x})$ and then $\|\boldsymbol{x}'\|_0 \geqslant k$.

Now, let $\boldsymbol{x}'$ satisfy $\|\boldsymbol{x}' - \boldsymbol{x}\| < \varepsilon$.

- If $\|\boldsymbol{x}'\|_0 = k$, then necessarily, $\mathcal{S}(\boldsymbol{x}') = \mathcal{S}(\boldsymbol{x}) = \mathcal{Q}$. By definition of $\boldsymbol{x} = \boldsymbol{x}_{\mathcal{Q}}$, $\mathcal{E}(\boldsymbol{x}') \geqslant \mathcal{E}(\boldsymbol{x})$. It follows that $\boldsymbol{x}$ is a local minimizer of the ℓ_0-constrained problem. Also, $\mathcal{J}(\boldsymbol{x}';\lambda) \geqslant \mathcal{J}(\boldsymbol{x};\lambda)$ holds.
- If $\|\boldsymbol{x}'\|_0 > k$, then $\mathcal{J}(\boldsymbol{x}';\lambda) = \mathcal{E}(\boldsymbol{x}') + \lambda\|\boldsymbol{x}'\|_0 \geqslant \mathcal{E}(\boldsymbol{x}') + \lambda(k+1)$. By continuity of $\mathcal{E}$, there exists a neighborhood $\mathcal{V}(\boldsymbol{x})$ of $\boldsymbol{x}$ such that if $\boldsymbol{x}' \in \mathcal{V}(\boldsymbol{x})$, $|\mathcal{E}(\boldsymbol{x}') - \mathcal{E}(\boldsymbol{x})| < \lambda$. Thus, if $\boldsymbol{x}' \in \mathcal{V}(\boldsymbol{x})$

and $\|\boldsymbol{x}' - \boldsymbol{x}\| < \varepsilon$, $\mathcal{J}(\boldsymbol{x}';\lambda) > \mathcal{E}(\boldsymbol{x}) + \lambda k = \mathcal{J}(\boldsymbol{x};\lambda)$.
Finally, if $\boldsymbol{x}' \in \mathcal{V}(\boldsymbol{x})$ and $\|\boldsymbol{x}' - \boldsymbol{x}\| < \varepsilon$, then $\mathcal{J}(\boldsymbol{x}';\lambda) > \mathcal{J}(\boldsymbol{x};\lambda)$. □

OLS as a special case : When $\lambda = 0$, SBR coincides with the well known Orthogonal Least Squares (OLS) algorithm [23], [42]. The removal operation never occurs, because it automatically leads to an increase of the squared error $\mathcal{K}_{\mathcal{Q}}(0) = \mathcal{E}_{\mathcal{Q}}$. Consequently, only insertions are worth being tested.

Empty solutions : We now characterize the λ-values for which SBR yields an empty solution.

Proposition 2. *The output of SBR($\emptyset;\lambda$) is the empty set* if and only if $\lambda \geqslant \lambda_{max}$, *with* $\lambda_{max} \triangleq \max_i(\langle \boldsymbol{a}_i, \boldsymbol{y}\rangle^2/\|\boldsymbol{a}_i\|^2)$.

Proof. SBR stops during its first iteration if all the insertion trials fail : $\forall i, \mathcal{E}_{\{i\}} + \lambda \geqslant \mathcal{E}_{\emptyset} = \|\boldsymbol{y}\|^2$. Given i, $\|\boldsymbol{y} - x_i\boldsymbol{a}_i\|^2$ is minimal when $x_i = \langle \boldsymbol{a}_i, \boldsymbol{y}\rangle/\|\boldsymbol{a}_i\|^2$, leading to $\mathcal{E}_{\{i\}} = \|\boldsymbol{y}\|^2 - \langle \boldsymbol{a}_i, \boldsymbol{y}\rangle^2/\|\boldsymbol{a}_i\|^2$. Thus, SBR stops during its first iteration *if and only if* $\forall i, \lambda \geqslant \langle \boldsymbol{a}_i, \boldsymbol{y}\rangle^2/\|\boldsymbol{a}_i\|^2$, *i.e.,* $\lambda \geqslant \lambda_{max}$. □

This result allows us to design an automatic procedure which sets a number of λ-values adaptively to the data in order to compute SBR solutions at different sparsity levels (see Section 2.6.4).

Reduced search : Instead of testing all the replacements $\mathcal{Q}' = \mathcal{Q} \bullet i$ at each SBR iteration, it is advantageous, if possible, to explore only a subset of these n replacements. We give two ideas to reduce the number of tests : the first is an acceleration of SBR, yielding the same iterates with a reduced search. The second idea is a modification of SBR.

Given an active set $\mathcal{Q}$, a removal $\mathcal{Q}' = \mathcal{Q}\backslash\{i\}$ yields an increase of the squared error and a decrease of the penalty equal to λ. Hence, the maximum decrease of the ℓ_0-penalized cost function which can be expected with a removal is $\lambda : \mathcal{K}_{\mathcal{Q}}(\lambda) - \mathcal{K}_{\mathcal{Q}'}(\lambda) \leqslant \lambda$. Consequently, if a given insertion $\mathcal{Q}' = \mathcal{Q} \cup \{i\}$ is such as $\mathcal{K}_{\mathcal{Q}}(\lambda) - \mathcal{K}_{\mathcal{Q}'}(\lambda) > \lambda$, then no removal is worth being tested. The acceleration of SBR thus consists in testing all the insertions first, and if the best insertion yields a decrease larger than λ, selecting the best insertion. Otherwise, all the removals have to be tested as stated in Tab. 2.1. This acceleration does not modify the SBR iterates. However, the gain is limited when the level of sparsity is high, *i.e.,* when the number of removals to be tested is reduced.

Haugland and Zhang pointed out that in a forward-backward strategy, it can be helpful to favor removals [35], [36]. Adapted to SBR, this idea leads to a modified algorithm in which removals are tested first, and the insertions are tested only if no removal yields a decrease of $\mathcal{K}_{\mathcal{Q}}(\lambda)$. If some removals decrease $\mathcal{K}_{\mathcal{Q}}(\lambda)$, then the removal yielding the maximal decrease is selected. In our experiments, the average qualitative performance of SBR and this modified version are quite comparable (there is no obvious gain or loss of quality nor a significant saving in computation time). Thus, in the following, we keep the version of SBR presented in Tab. 2.1.

2.4 Implementation issues

Given the current active set $\mathcal{Q}$, an SBR iteration consists in computing the squared error $\mathcal{E}_{\mathcal{Q}'}$ for any replacement $\mathcal{Q}' = \mathcal{Q} \bullet i$, leading to the computation of $\mathcal{K}_{\mathcal{Q}'}(\lambda) = \mathcal{E}_{\mathcal{Q}'} + \lambda \mathrm{Card}\,[\mathcal{Q}']$. We first describe a basic implementation in which $\mathcal{E}_{\mathcal{Q}'}$ is computed independently of the knowledge of $\mathcal{E}_{\mathcal{Q}}$. Then, we present an efficient implementation allowing a fast update when $\mathcal{Q}$ is modified. We will denote by $k \triangleq \mathrm{Card}\,[\mathcal{Q}]$ the cardinality of the active set ($k \leqslant \min(m, n)$).

2.4.1 Basic implementation

The minimization problem (2.1) reduces to the unconstrained minimization of $\|\boldsymbol{y}-\boldsymbol{A}_{\mathcal{Q}}\boldsymbol{t}\|^2$ w.r.t. $\boldsymbol{t}\in\mathbb{R}^k$. Because $\boldsymbol{A}_{\mathcal{Q}}$ is full rank, this problem has a unique minimizer that reads :

$$\boldsymbol{t}_{\mathcal{Q}} \triangleq \arg\min_{\boldsymbol{t}} \|\boldsymbol{y}-\boldsymbol{A}_{\mathcal{Q}}\boldsymbol{t}\|^2 = (\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{A}_{\mathcal{Q}})^{-1}\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{y} \tag{2.13}$$

and the minimal squared error reads :

$$\mathcal{E}_{\mathcal{Q}} = \|\boldsymbol{y}-\boldsymbol{A}_{\mathcal{Q}}\boldsymbol{t}_{\mathcal{Q}}\|^2 = \|\boldsymbol{y}\|^2 - \boldsymbol{y}^t\boldsymbol{A}_{\mathcal{Q}}\boldsymbol{t}_{\mathcal{Q}} \tag{2.14}$$

Finally, given the active set $\mathcal{Q}$, an SBR iteration involves the computation of $\boldsymbol{t}_{\mathcal{Q}'}$ and $\mathcal{E}_{\mathcal{Q}'}$ for all possible replacements $\mathcal{Q}' = \mathcal{Q}\bullet i$, using (2.13) and (2.14).

2.4.2 Recursive implementation

At each SBR iteration, n least-square problems of the form (2.13) must be solved, each requiring the inversion of the Gram matrix $\boldsymbol{G}_{\mathcal{Q}} \triangleq \boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{A}_{\mathcal{Q}}$ of size $k\times k$. The computation cost can be high since in the general case, a matrix inversion costs $\mathcal{O}(k^3)$ scalar operations. Following an idea widely spread in the subset selection literature, we propose to solve (2.13) in a recursive manner.

A first possibility is to use the Gram-Schmidt procedure [23], [42] which yields an orthogonal decomposition of $\boldsymbol{A}_{\mathcal{Q}} = \boldsymbol{W}\boldsymbol{U}$, where $\boldsymbol{W}$ is an $m\times k$ matrix with orthogonal columns and $\boldsymbol{U}$ is a $k\times k$ upper triangular matrix. Although it yields an efficient updating strategy when including an index into the active set (leading to the update of $\boldsymbol{A}_{\mathcal{Q}'} = [\boldsymbol{A}_{\mathcal{Q}}, \boldsymbol{a}_i]$), the Gram-Schmidt procedure does not extend with the same level of efficiency when an index removal is considered [34].

An alternative is to use the block matrix inversion lemma [43] allowing an efficient update of $\boldsymbol{G}_{\mathcal{Q}}^{-1}$ for both index insertion and removal. An efficient SMLR implementation is proposed in [13], based on the recursive update of matrices of the form $(\boldsymbol{G}_{\mathcal{Q}}+\alpha\boldsymbol{I}_k)^{-1}$. This approach can easily be adapted to SBR where the matrix to update is $\boldsymbol{G}_{\mathcal{Q}}^{-1}$ (see also [44] for the downdate step). However, we have observed numerical instabilities when the selected columns of $\boldsymbol{A}$ are highly correlated.

A more stable solution is based on the Cholesky factorization $\boldsymbol{G}_{\mathcal{Q}} = \boldsymbol{L}_{\mathcal{Q}}\boldsymbol{L}_{\mathcal{Q}}^t$, where $\boldsymbol{L}_{\mathcal{Q}}$ is a lower triangular matrix. Updating $\boldsymbol{L}_{\mathcal{Q}}$ is advantageous since it is better conditioned than $\boldsymbol{G}_{\mathcal{Q}}^{-1}$. This update can be easily done in the insertion case [45]. It is less easy for removals, since they break the triangular structure of $\boldsymbol{L}_{\mathcal{Q}}$. A recursive update of the Cholesky factor of $\boldsymbol{G}_{\mathcal{Q}}^{-1}$ was recently proposed [46]. Here, we introduce a simpler strategy relying on the factorization of $\boldsymbol{G}_{\mathcal{Q}}$.

2.4.3 Efficient strategy based on the Cholesky factorization

First, we notice that a new column $\boldsymbol{a}_i$ can be inserted at the last position in $\boldsymbol{A}_{\mathcal{Q}\cup\{i\}}$ to compute the value of $\mathcal{E}_{\mathcal{Q}\cup\{i\}}$. On the contrary, when removing a column, we do not know *a priori* the position of the column to be removed, thus it cannot be assumed to be the last column of $\boldsymbol{A}_{\mathcal{Q}}$. We will hence describe the cases where :

- a non active element $i\notin\mathcal{Q}$ is included after the other columns : $\boldsymbol{A}_{\mathcal{Q}'} = [\boldsymbol{A}_{\mathcal{Q}}, \boldsymbol{a}_i]$;
- an active element $i\in\mathcal{Q}$ is to be removed, the column $\boldsymbol{a}_i$ being in an arbitrary position.

$\boldsymbol{G}_{\mathcal{Q}}$ being a symmetric positive-definite matrix, it reads $\boldsymbol{G}_{\mathcal{Q}} = \boldsymbol{L}_{\mathcal{Q}}\boldsymbol{L}_{\mathcal{Q}}^t$ where the Cholesky factor $\boldsymbol{L}_{\mathcal{Q}}$ is a lower triangular matrix of size $k\times k$. Applying (2.13), the least-square minimizer rereads $\boldsymbol{t}_{\mathcal{Q}} = \boldsymbol{L}_{\mathcal{Q}}^{-t}\boldsymbol{L}_{\mathcal{Q}}^{-1}\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{y}$ where the superscript $-t$ refers to the inverse transposition operator, and (2.14) yields :

$$\mathcal{K}_{\mathcal{Q}}(\lambda) = \mathcal{E}_{\mathcal{Q}}(\lambda) + \lambda k = \|\boldsymbol{y}\|^2 - \|\boldsymbol{L}_{\mathcal{Q}}^{-1}\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{y}\|^2 + \lambda k \tag{2.15}$$

Given $\boldsymbol{L}_{\mathcal{Q}}$, $\mathcal{O}(k^2)$ scalar operations are required to solve the triangular system $\boldsymbol{L}_{\mathcal{Q}}^{-1}(\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{y})$.

Insertion of a new column after the existing columns : Including a new column leads to $\boldsymbol{A}_{\mathcal{Q}'} = [\boldsymbol{A}_{\mathcal{Q}}, \boldsymbol{a}_i]$. Thus, the new Gram matrix can be expressed as a 2×2 block matrix :

$$\boldsymbol{G}_{\mathcal{Q}'} = \begin{bmatrix} \boldsymbol{G}_{\mathcal{Q}} & \boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{a}_i \\ (\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{a}_i)^t & \|\boldsymbol{a}_i\|^2 \end{bmatrix} \tag{2.16}$$

and the Cholesky factor of $\boldsymbol{G}_{\mathcal{Q}'}$ can be straightforwardly updated :

$$\boldsymbol{L}_{\mathcal{Q}'} = \begin{bmatrix} \boldsymbol{L}_{\mathcal{Q}} & \boldsymbol{0} \\ \boldsymbol{l}_{\mathcal{Q},i}^t & \alpha_{\mathcal{Q},i} \end{bmatrix} \tag{2.17}$$

with $\boldsymbol{l}_{\mathcal{Q},i} = \boldsymbol{L}_{\mathcal{Q}}^{-1}\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{a}_i$ and $\alpha_{\mathcal{Q},i} = (\|\boldsymbol{a}_i\|^2 - \|\boldsymbol{l}_{\mathcal{Q},i}\|^2)^{1/2}$.

The computation of $\mathcal{K}_{\mathcal{Q}'}(\lambda)$ using (2.15) requires two triangular system inversions (computation of $\boldsymbol{l}_{\mathcal{Q},i}$ and computation of $\mathcal{K}_{\mathcal{Q}'}(\lambda)$). However, by computing

$$\mathcal{K}_{\mathcal{Q}'}(\lambda) - \mathcal{K}_{\mathcal{Q}}(\lambda) = \lambda - (\boldsymbol{l}_{\mathcal{Q},i}^t\boldsymbol{L}_{\mathcal{Q}}^{-1}\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{y})^2/\alpha_{\mathcal{Q},i}^2 \tag{2.18}$$

the cost can be reduced up to the pre-computation of $\boldsymbol{L}_{\mathcal{Q}}^{-1}(\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{y})$ at the beginning of the SBR iteration. The computation of $\mathcal{K}_{\mathcal{Q}'}(\lambda)$ only requires one triangular system inversion (computation of $\boldsymbol{l}_{\mathcal{Q},i}$).

Removal of an arbitrary column : When removing a column $\boldsymbol{a}_i$, updating $\boldsymbol{L}_{\mathcal{Q}}$ remains possible although slightly more expensive. This idea was first developed by Ge *et al.* [46] who update the Cholesky factorization of matrix $\boldsymbol{G}_{\mathcal{Q}}^{-1}$. We adapt it to the direct (simpler) factorization of $\boldsymbol{G}_{\mathcal{Q}}$. Let I be the index such that $\boldsymbol{a}_i$ is the I-th column of $\boldsymbol{A}_{\mathcal{Q}}$ (with $1 \leqslant I \leqslant k$). $\boldsymbol{L}_{\mathcal{Q}}$ can be written in a block matrix form :

$$\boldsymbol{L}_{\mathcal{Q}} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{b}^t & d & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{e} & \boldsymbol{F} \end{bmatrix}. \tag{2.19}$$

where the lowercase characters refer to the scalar (d) and vector quantities ($\boldsymbol{b}$,$\boldsymbol{e}$) which appear in the I-th row and in the I-th column. The computation of $\boldsymbol{G}_{\mathcal{Q}} = \boldsymbol{L}_{\mathcal{Q}}\boldsymbol{L}_{\mathcal{Q}}^t$ and the removal of the I-th row and the I-th column in $\boldsymbol{G}_{\mathcal{Q}}$ leads to

$$\boldsymbol{G}_{\mathcal{Q}'} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{F} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}^t & \boldsymbol{C}^t \\ \boldsymbol{0} & \boldsymbol{F}^t \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{e} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{e}^t \end{bmatrix} \tag{2.20}$$

By identification of this expression with the Cholesky factorization $\boldsymbol{G}_{\mathcal{Q}'} = \boldsymbol{L}_{\mathcal{Q}'}\boldsymbol{L}_{\mathcal{Q}'}^t$ and because the Cholesky factorization is unique, $\boldsymbol{L}_{\mathcal{Q}'}$ necessarily reads :

$$\boldsymbol{L}_{\mathcal{Q}'} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{X} \end{bmatrix} \tag{2.21}$$

where $\boldsymbol{X}$ is a lower triangular matrix satisfying $\boldsymbol{X}\boldsymbol{X}^t = \boldsymbol{F}\boldsymbol{F}^t + \boldsymbol{e}\boldsymbol{e}^t$. The problem of computing $\boldsymbol{X}$ from $\boldsymbol{F}$ and $\boldsymbol{e}$ is classical; it is known as a positive rank 1 Cholesky update and there exists a stable algorithm in $\mathcal{O}(f^2)$ operations, where $f = k - I$ is the size of $\boldsymbol{F}$ [47].

Finally, the computation of $\mathcal{K}_{\mathcal{Q}'}(\lambda)$ involves a positive Cholesky update and a triangular system inversion in (2.15). Thus, its overall cost is in $\mathcal{O}(k^2)$. Notice that matrix $\boldsymbol{F}$ is of size $k-I$. Therefore, the cost of a Cholesky update completely depends on the position I of the column $\boldsymbol{a}_i$ to be removed. The larger I, the less expensive is the Cholesky update.

TABLE 2.2 – Efficient implementation of an SBR iteration

Input : $\mathcal{Q}$, λ
Pre-computed quantities : $\boldsymbol{A}^t\boldsymbol{y}$ and $\|\boldsymbol{a}_i\|^2$ for all i
Stored quantities : $\mathcal{K}_{\mathcal{Q}}(\lambda)$, $\boldsymbol{L}_{\mathcal{Q}}$, and $\boldsymbol{L}_{\mathcal{Q}}^{-1}(\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{y})$
Set $\ell = 0$,`least_cost=`$\mathcal{K}_{\mathcal{Q}}(\lambda)$.
For $i = 1$ to n,
If $i \notin \mathcal{Q}$, `/*Insertion test*/`
Compute $\boldsymbol{l}_{\mathcal{Q},i} = \boldsymbol{L}_{\mathcal{Q}}^{-1}\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{a}_i$, and $\mathcal{K}_{\mathcal{Q}'}(\lambda)$ using (2.18).
else, `/*Removal test*/`
Update the Cholesky decomposition : $\boldsymbol{X}$=`cholupdate(`$\boldsymbol{F}$,$\boldsymbol{e}$,+).
Compute $\boldsymbol{L}_{\mathcal{Q}'}$ and $\mathcal{K}_{\mathcal{Q}'}(\lambda)$ using (2.21) and (2.15).
End if.
If $\mathcal{K}_{\mathcal{Q}'}(\lambda)$ <`least_cost`,
Set $\ell = i$, `least_cost=`$\mathcal{K}_{\mathcal{Q}'}(\lambda)$.
End if.
End for.
If $\ell \neq 0$, `/*Perform the single replacement*/`
Set $\mathcal{Q}' = \mathcal{Q} \bullet \ell, \mathcal{K}_{\mathcal{Q}'}(\lambda) =$`least_cost`.
Compute $\boldsymbol{L}_{\mathcal{Q}'}$ using (2.17) or (2.21), and then $\boldsymbol{L}_{\mathcal{Q}'}^{-1}(\boldsymbol{A}_{\mathcal{Q}'}^t\boldsymbol{y})$.
else,
Terminate SBR.
End if.
Output : next iterate $\mathcal{Q}' = \mathcal{Q} \bullet \ell, \mathcal{K}_{\mathcal{Q}'}(\lambda)$, $\boldsymbol{L}_{\mathcal{Q}'}$ and $\boldsymbol{L}_{\mathcal{Q}'}^{-1}(\boldsymbol{A}_{\mathcal{Q}'}^t\boldsymbol{y})$

2.4.4 Memory requirements and computation burden

The efficient (fast and stable) procedure is summarized in Tab. 2.2. Given the current active set $\mathcal{Q}$, the index ℓ defining the next SBR iterate $\mathcal{Q} \bullet \ell$ is chosen according to (2.11) and $\boldsymbol{L}_{\mathcal{Q}\bullet\ell}$ is finally updated. No update of the amplitudes is necessary. The actual implementation may vary depending on the size and the structure of matrix $\boldsymbol{A}$. We briefly detail the main possible implementations and their requirements in terms of storage and computation. Regarding the computation burden, we count the number of elementary operations expressed in terms of scalar multiplications since the cost of a scalar addition is negligible with respect to that of a multiplication.

When $\boldsymbol{A}$ is relatively small, the computation and storage of the full Gram matrix $\boldsymbol{A}^t\boldsymbol{A}$ prior to any SBR iteration (storage of n^2 scalar elements) avoids to recompute the vectors $\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{a}_i$ which are needed when the insertion of $\boldsymbol{a}_i$ into the active set is tested. Similarly, we store $\boldsymbol{A}^t\boldsymbol{y}$ and the values $\|\boldsymbol{a}_i\|^2$ in two 1D arrays of size n, prior to any SBR loop. The storage of the other quantities (mainly $\boldsymbol{L}_{\mathcal{Q}}$) that are being updated amounts to $\mathcal{O}(k^2)$ scalar elements and each test costs $\mathcal{O}(k^2)$ elementary operations, as it involves the inversion of a triangular system of size $k \times k$, plus a positive rank 1 Cholesky update in the removal case. This cost has to be compared with the $\mathcal{O}(k^3)$ scalar operations which are necessary when inverting the Gram matrix in the basic implementation of SBR.

When $\boldsymbol{A}$ is larger, the storage of $\boldsymbol{A}^t\boldsymbol{A}$ is no longer possible and vectors $\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{a}_i$ must be recomputed at any SBR iteration, for each insertion test $\mathcal{Q}' = \mathcal{Q} \cup \{i\}$. The computation of $\boldsymbol{A}_{\mathcal{Q}}^t\boldsymbol{a}_i$ costs km elementary operations and represents the most important cost of an insertion test. Indeed, the remaining part is in $\mathcal{O}(k^2)$ and for sparse representations, k is expected to be much lower than m. The cost of a single replacement finally amounts to $\mathcal{O}(k^2) + \mathcal{O}(km)$ elementary operations.

When the dictionary has some specific structure, the above storage limitation can be alleviated, enabling a fast implementation even when n is large. For instance, if a large number of pairs of columns of $\boldsymbol{A}$ are orthogonal to each other, $\boldsymbol{A}^t\boldsymbol{A}$ can be stored as a sparse array. Also, finite

TABLE 2.3 – Separation of two Gaussian features from noise-free data with SBR. d stands for the distance between the Gaussian features. We display the size of the support $\mathcal{Q}(\lambda; d)$ obtained with a sequence of decreasing λ-values $\lambda_0 > \lambda_1 > \ldots > \lambda_7$. The label * indicates an exact recovery for a support of cardinality 2

λ	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	$\lambda \leqslant \lambda_7$
$d = 20$	0	0	2^*	2^*	2^*	2^*	2^*	2^*
$d = 13$	0	1	3	4	5	2^*	2^*	2^*
$d = 6$	0	1	1	3	5	6	8	2^*

impulse response deconvolution problems enable a fast implementation since $\boldsymbol{A}^t\boldsymbol{A}$ is then a Toeplitz matrix (save north-west and/or south-east submatrices, depending on the boundary conditions). The knowledge of the auto-correlation of the impulse response is sufficient to describe most of the Gram matrix.

All these variants have been implemented (Matlab codes are available to academic users from the authors upon request). In the following, we analyze the behavior of SBR on two difficult problems, in which the dictionaries are highly correlated : the deconvolution of a sparse signal with a Gaussian impulse response (Section 2.5) and the joint detection of discontinuities at different orders in a signal (Section 2.6).

2.5 Deconvolution of a sparse signal with a Gaussian impulse response

This is a typical problem for which SMLR was introduced [13]. It affords us to study the ability of SBR to perform an exact recovery in a simple noise-free case (separation of two Gaussian features from noise-free data) and to test the behavior of SBR in a noisy case (estimation of a larger number of Gaussian features). For simulated problems, we denote by $\boldsymbol{x}^*$ the exact sparse signal and we generate noisy data according to $\boldsymbol{y} = \boldsymbol{y}^* + \boldsymbol{n} = \boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{n}$, where $\boldsymbol{y}^* = \boldsymbol{A}\boldsymbol{x}^*$ denotes the noise-free data and $\boldsymbol{n}$ stands for the observation noise. The dictionary columns $\boldsymbol{a}_i$ are always normalized : $\|\boldsymbol{a}_i\|^2 = 1$. The signal to noise ratio (SNR) is defined by SNR=$10\log(P_Y/P_N)$, where $P_Y = \|\boldsymbol{y}^*\|^2/m$ is the average power of the noise-free data and P_N is the variance of the noise process $\boldsymbol{n}$.

2.5.1 Dictionary and simulated data

The impulse response $\boldsymbol{h}$ is a Gaussian signal of standard deviation σ, sampled on a regular grid at integer locations. It is approximated by a finite impulse response of length 6σ by thresholding the smallest values, allowing a fast implementation even for large size problems (see subsection 2.4.4). The deconvolution problem leads to a Toeplitz matrix $\boldsymbol{A}$ whose columns $\boldsymbol{a}_i$ are obtained by shifting the signal $\boldsymbol{h}$. The dimension of $\boldsymbol{A}$ is chosen to have any Gaussian feature resulting from the convolution $\boldsymbol{h} * \boldsymbol{x}^*$ belonging to the observation window $\{1, \ldots, m\}$. This implies that $\boldsymbol{A}$ is slightly undercomplete ($m > n$). Denoting by $n_h = 1 + 2\text{round}(3\sigma)$ the size of the support of $\boldsymbol{h}$, the data size reads $m = n + n_h - 1$.

2.5.2 Separation of two close Gaussian features

We first analyze the ability of SBR to separate two Gaussian features ($\|\boldsymbol{x}^*\|_0 = 2$) from noise-free data. The centers of both Gaussian features lay at a relative distance d (expressed as a number of samples) and their weights x_i^* are set to 1. We generate the corresponding noise-free data $\boldsymbol{y}^*$ and

we run SBR($\emptyset; \lambda$) with a number of predefined λ-values. We analyze the SBR outputs $\mathcal{Q}(\lambda, d)$ by computing their size Card $[\mathcal{Q}(\lambda; d)]$ and by testing if $\mathcal{Q}(\lambda; d)$ is equal to the true support $\mathcal{S}(\boldsymbol{x}^*)$. Tab. 2.3 shows the results obtained for a problem of size 300×270 ($m = 300$, σ=5, and n_h=31) with distances equal to $d = 20, 13$, and 6 samples. The grid of λ-values for which SBR is run is common to the three tests. The maximal value λ_0 is chosen in such a way that the output $\mathcal{Q}(\lambda_0; d)$ is empty (see Proposition 2) and the other values are set according to $\lambda_j = \lambda_0/10^j$. It is noticeable that the exact recovery is always reached provided that λ is sufficiently small. This result remains true even for smaller distances (from $d = 2$). When the Gaussian features strongly overlap, *i.e.,* for $d \leqslant 13$, the size of the support obtained as output first increases while λ decreases, and then for lower λ-values, removals start to occur, enabling the exact recovery. Similarly to SBR, forward algorithms such as OMP and OLS start by positioning a (wrong) Gaussian feature in between the two Gaussians in their first iteration but in the latter case, the early wrong detection disables an exact recovery.

2.5.3 Behavior of SBR for noisy data

We run SBR on more realistic noisy data and on a larger problem ($m = 3000$ samples). The unknown sparse signal $\boldsymbol{x}^*$ is composed of 17 Gaussian features. The impulse response $\boldsymbol{h}$ is of size $n_h = 301$ ($\sigma = 50$) yielding an observation matrix $\boldsymbol{A}$ of size 3000×2700, and the SNR is set to 20 dB.

Fig. 2.1 displays the simulated data and the SBR results obtained with a few λ-values. When λ decreases, the SBR approximations are of better quality but less sparse. For large λ-values, only the main Gaussian features are found, and then, when λ decreases, the smaller features are being recovered together with unwanted features. Removals rarely occur for coarse approximations. They occur more frequently when two estimated features are overlapping and for low λ-values. On the simulation of Fig. 2.1, removals occur for $\lambda \leqslant 0.15$, yielding approximations that are more accurate than those obtained with OLS and for the same cardinality (the residual $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ is lower), while when $\lambda > 0.15$, the SBR output coincides with the OLS iterate of same cardinality. Although the performance of SBR is at least equal to that of OLS, the exact support of $\boldsymbol{x}^*$ is never found. However, it must be stressed that the problem is very difficult because the data are noisy and the neighboring columns of $\boldsymbol{A}$ are highly correlated. In such difficult case, one needs to perform a wider exploration of the discrete set $\{0, 1\}^n$ by introducing moves that are more complex than single replacements. Such extensions were already proposed in the case of SMLR. One can for instance shift a detected spike x_i forwards of backwards [48] or update a block of neighboring components jointly (*e.g.,* x_i and x_{i+1}) [49].

2.6 Joint detection of discontinuities at different orders in a signal

We now consider another challenging problem, the joint detection of discontinuities at different orders in a signal. We process both simulated and real data and compare the performance of SBR with respect to other sparse approximation algorithms (OMP and OLS) in terms of discontinuity estimation, approximation accuracy, and computation time. Firstly, we formulate the detection of discontinuities at a single order p as a spline approximation problem. Then, we take advantage of this formulation to introduce the joint detection problem more easily.

2.6.1 Approximation of a spline of degree p

In the continuous case, a signal is a spline of degree p with k knots *if and only if* its $(p + 1)$-th derivative is a stream of k weighted Diracs [50]. In the discrete case, we introduce the dictionary $\boldsymbol{A}^p$

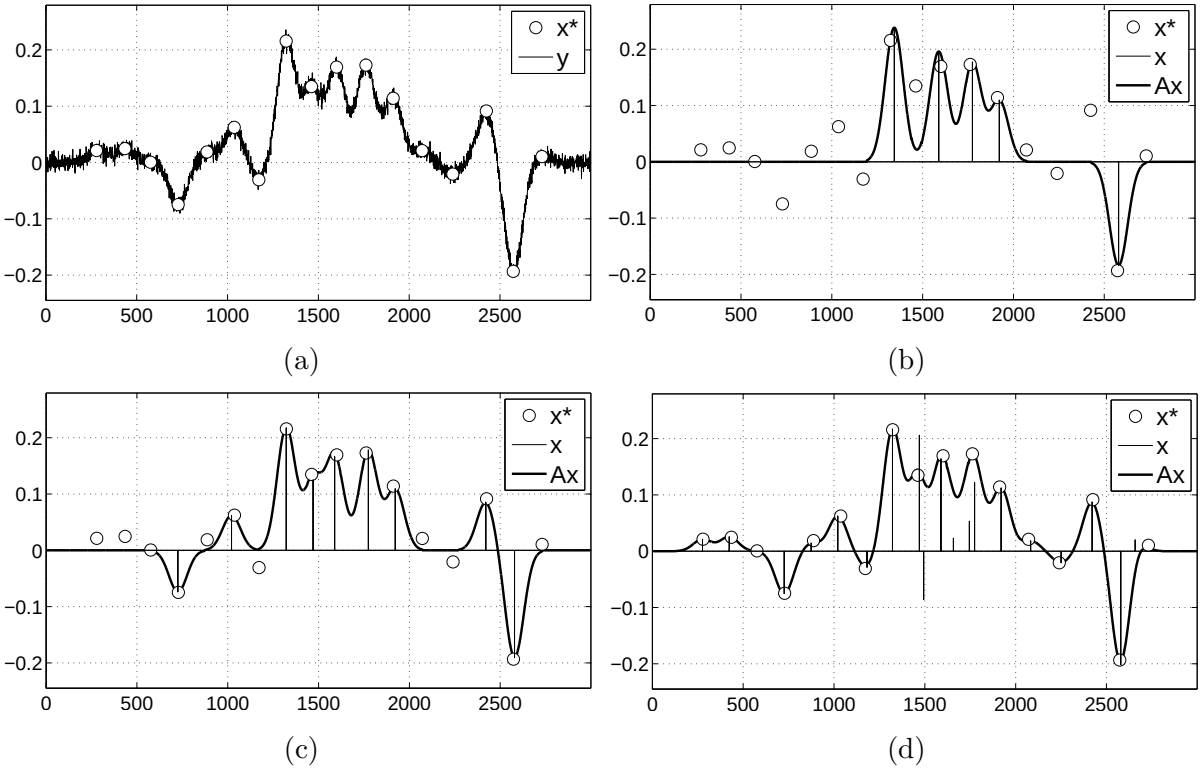


FIGURE 2.1 – Gaussian deconvolution results. Problem of size 3000×2700, ($\sigma = 50$). (a) Generated data, with 17 Gaussian features and with SNR=20 dB. The exact locations $\boldsymbol{x}^*$ are labeled o. (b,c,d) SBR outputs and data approximations with empirical settings of λ. The estimated amplitudes $\boldsymbol{x}$ are shown with vertical spikes. The SBR outputs (supports) are of size 5, 9, and 19, respectively. The computation time always remains below 8 seconds (Matlab implementation).

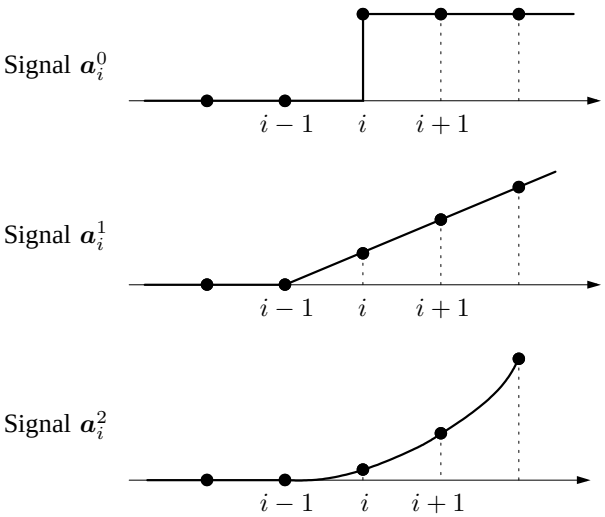
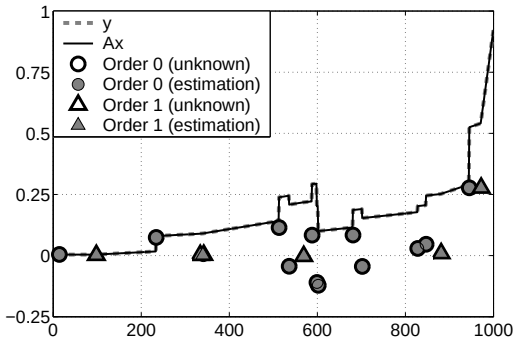
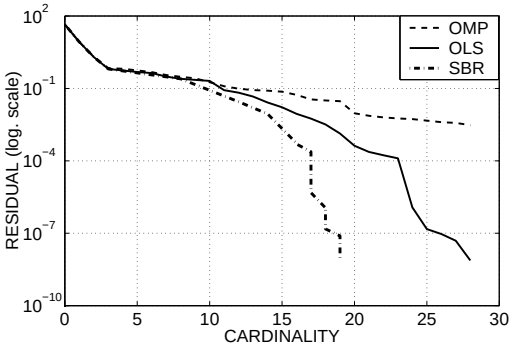


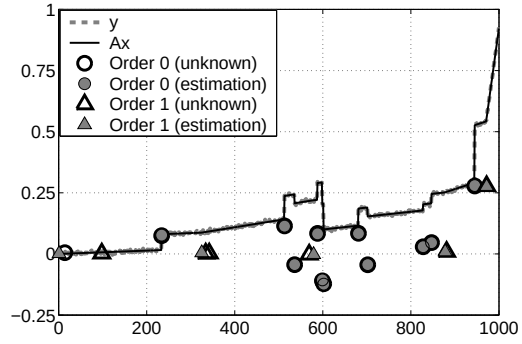
FIGURE 2.2 – Signals $\boldsymbol{a}_i^p$ related to the p-th order discontinuities at location i. $\boldsymbol{a}_i^0$ is the Heaviside step function, $\boldsymbol{a}_i^1$ is the ramp function, and $\boldsymbol{a}_i^2$ is the one-sided quadratic function. Each signal is equal to 1 at location i and its support is equal to $\{i, \ldots, m\}$.



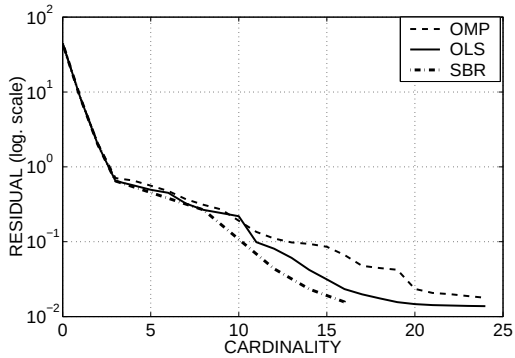
(a) Noise-free data and SBR approximation



(b) "ℓ_2-ℓ_0" curves (noise-free data)



(c) Noisy data (SNR = 35 dB) and SBR approximation



(d) "ℓ_2-ℓ_0" curves (noisy data)

FIGURE 2.3 – Joint detection of discontinuities at orders 0 and 1. The dictionary is of size 1000×1999 and the data signal $\boldsymbol{y}$ includes 18 discontinuities. The true and estimated discontinuity locations are represented with unfilled black and filled gray labels. The shape of the labels (circular or triangular) indicates the discontinuity order. The dashed gray and solid black curves represent the data signal $\boldsymbol{y}$ and its approximation $\boldsymbol{Ax}$ for the least λ-value. (a) Signal approximation from noise-free data. The recovery is exact and both curves are superimposed. (b) "ℓ_2-ℓ_0" curves showing the squared residual versus the cardinality for SBR, OLS, and OMP. The SBR performance is expressed only for the λ-values that are larger than λ_{20}, because below this value, the recovery is exact and the log-residual is equal to $-\infty$. (c,d) Similar results for noisy data (SNR = 35 dB).

formed of shifted versions of the one-sided power function $k \mapsto k_+^p \triangleq [\max(k,0)]^p$ for all possible shifts (see Fig. 2.2). $\boldsymbol{A}^p$ represents the integration operator of degree $p+1$. Denoting by $\{1,\ldots,m\}$ the support of the data signal $\boldsymbol{y}$, the shifted signals $\boldsymbol{a}_i^p$ (for $i \in \{1,\ldots,m\}$) read

$$\forall k \in \{1,\ldots,,m\}, \boldsymbol{a}_i^p(k) = (k-i+1)_+^p \tag{2.22}$$

and their support is equal to $\{i,\ldots,m\}$. Finally, we form the dictionary $\boldsymbol{A}^p = [\boldsymbol{a}_1^p,\ldots,\boldsymbol{a}_{m-p}^p]$ of size $m \times (m-p)$. It does not make sense to allow the occurrence of a p-th order discontinuity for the last samples (*i.e.,*, to include $\boldsymbol{a}_i^p$ for $i > m-p$) since the spline approximation would require to reconstruct a polynomial of degree p in the range $\{i,\ldots,m\}$ from less than $p+1$ data samples.

We address the spline approximation problem as the sparse approximation of $\boldsymbol{y}$ by the piecewise polynomial $\boldsymbol{g}^p = \boldsymbol{A}^p\boldsymbol{x}^p$ (actually, we impose as initial condition that the spline function is equal to 0 for $k \leqslant 0$). The sparse approximation consists in the detection of the discontinuity locations (also referred to as knots in the spline approximation literature) and the estimation of their amplitudes : x_i^p codes for the amplitude of a jump at location $i(p=0)$, the change of slope at location $i(p=1)$, *etc.* Here, the notion of sparsity is related to the number of discontinuity locations.

2.6.2 Approximation of a piecewise polynomial of maximum degree P

Following [50], we formulate this problem as the joint detection of discontinuities at orders $p = 0, \dots, P$. Let us append the elementary dictionaries $\boldsymbol{A}^p$ in a global dictionary $\boldsymbol{A} = [\boldsymbol{A}^0, \dots, \boldsymbol{A}^P]$. The approximation $\boldsymbol{g} = \boldsymbol{A}\boldsymbol{x}$ of a given signal rereads $\boldsymbol{g} = \sum_p \boldsymbol{A}^p \boldsymbol{x}^p$ where vector $\boldsymbol{x} = \{\boldsymbol{x}^0, \dots, \boldsymbol{x}^P\}$ gathers the p-th order amplitudes $\boldsymbol{x}^p$ for all p. When $\boldsymbol{x}$ is sparse, all vectors $\boldsymbol{x}^p$ are sparse and the approximation signal $\boldsymbol{g}$ is the sum of piecewise polynomials of degree lower than P with a limited number of pieces.

The dictionary $\boldsymbol{A}$ is overcomplete since it is of size $m \times s$, with $s = (P+1)(m - P/2) > m$ for all $P \geqslant 1$. Moreover, it is highly correlated : any column $\boldsymbol{a}_i^p$ is strongly correlated with all other columns $\boldsymbol{a}_j^q$ because their respective supports are the intervals $\{i, \dots, m\}$ and $\{j, \dots, m\}$, and hence overlap. The discontinuity detection problem is difficult, as most algorithms are very likely to position wrong discontinuities in their first iterations. For example, when approximating a signal with two discontinuities at distinct locations i and j, they start to position a first (wrong) discontinuity in between i and j, and forward algorithms cannot remove it (see Section 2.6.5 and Fig. 2.5 for details).

2.6.3 Adaptation of SBR

It is important to notice that the dictionary defined above does not satisfy the unique representation property. For instance, the difference between two discrete ramps at locations i and $i+1$ yields the discrete Heaviside function at location i : $\boldsymbol{a}_i^1 - \boldsymbol{a}_{i+1}^1 = \boldsymbol{a}_i^0$. More generally, for $p > 1$, $\boldsymbol{a}_i^p - \boldsymbol{a}_{i+1}^p$ reads as a linear combination of $\boldsymbol{a}_i^0$ and $\boldsymbol{a}_{i+1}^q, (q = 1, \dots, p-1)$.

As mentioned in Section 2.2, the SBR algorithm basically requires that the dictionary satisfies the URP to ensure that the Gram matrix $\boldsymbol{G}_\mathcal{Q} = \boldsymbol{A}_\mathcal{Q}^t \boldsymbol{A}_\mathcal{Q}$ is invertible, but this assumption can be relaxed provided that only full rank matrices $\boldsymbol{A}_\mathcal{Q}$ are explored. Here, SBR is slightly modified, based on the following proposition which gives a sufficient condition of invertibility of $\boldsymbol{G}_\mathcal{Q}$.

Lemma 1. *Consider an active set $\mathcal{Q}$ satisfying the condition of Proposition 3, and let $i^- = \min\{i | n_i > 0\}$ denote the lowest location of an active entry. Up to a reordering of the columns, $\boldsymbol{A}_\mathcal{Q}$ rereads $\boldsymbol{A}_\mathcal{Q} = [\boldsymbol{A}_{i^-}, \boldsymbol{A}_{\mathcal{Q}\setminus\{i^-\}}]$ with obvious notations. If $\boldsymbol{A}_{\mathcal{Q}\setminus\{i^-\}}$ is full rank, then $\boldsymbol{A}_\mathcal{Q}$ is also full rank.*

Proof. Let $I = n_{i^-}$ denote the number of discontinuities at location i^- and let $0 \leqslant p_1 < p_2 < \ldots < p_I$ denote their order, sorted in the ascending order. Suppose that there exist two families of scalars $\{\mu_{i^-}^{p_1}, \dots, \mu_{i^-}^{p_I}\}$ and $\{\mu_i^p | i \neq i^- \text{ and } i \text{ is active at order } p\}$ such that

$$\sum_{j=1}^{I} \mu_{i^-}^{p_j} \boldsymbol{a}_{i^-}^{p_j} + \sum_{i \neq i^-} \sum_p \mu_i^p \boldsymbol{a}_i^p = \boldsymbol{0} \tag{2.23}$$

Let us show that all μ-values are then equal to 0.

Rewriting the first I nonzero equations in this system and because $\mathcal{Q}$ satisfies the condition of Proposition 3, we have, for all $k \in \{i^-, \dots, i^- + I - 1\}$, $\sum_{j=1}^{I} \mu_{i^-}^{p_j} (k + i^- - 1)^{p_j} = 0$. In other words, the polynomial $F(X) = \sum_{j=1}^{I} \mu_{i^-}^{p_j} X^{p_j}$ has I positive roots. It is shown in [51, p. 76] that a non-zero polynomial formed of I monomials of different degree has at most $I-1$ positive roots. Therefore, F is the zero polynomial and all scalars $\mu_{i^-}^{p_j}$ are 0. We deduce from (2.23) and from the full rankness of $\boldsymbol{A}_{\mathcal{Q}\setminus\{i^-\}}$ that $\mu_i^p = 0$ for all (i, p).

We have shown that the column vectors of $\boldsymbol{A}_\mathcal{Q}$ are linearly independent, *i.e.,* that $\boldsymbol{A}_\mathcal{Q}$ is full

TABLE 2.4 – Behavior of the SBR iterates for both approximations of Fig. 2.5(a,d). The tables display the squared error $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ versus the cardinality $\|\boldsymbol{x}\|_0$ for each iterate. i stands for the iteration index.

SBR (λ=2085,4+/2-)

i	$\|\boldsymbol{x}\|_0$	Error
0	0	2101.408
1	1	16.870
2	**2**	**12.266**
3	3	3.074
4	4	642
5	3	663
6	**2**	**938**

SBR (λ=66,7+/2-)

i	$\|\boldsymbol{x}\|_0$	Error
0	0	2101.408
1	1	16.870
2	2	12.266
3	3	3.074
4	4	642
5	3	663
6	4	555
7	**5**	**480**
8	4	532
9	**5**	**464**

rank. □

Lemma 1 is a key element to prove following proposition.

Proposition 3. *Let n_i denote the number of discontinuities $\boldsymbol{a}_i^p, p = 0, \ldots, P$ which are being activated at sample i, i.e., for which $\boldsymbol{x}_i^p \neq 0$. Let us define the binary condition $\mathcal{C}(i)$:*

- *if $n_i = 0, \mathcal{C}(i) \triangleq 1$;*
- *if $n_i \geqslant 1, \mathcal{C}(i) \triangleq 1(\forall j \in \{1, \ldots, n_i - 1\}, n_{i+j} = 0)$.*

If Q is such that for all $i, \mathcal{C}(i) = 1$, then $\boldsymbol{G}_{\mathcal{Q}}$ is invertible.

The proof of Proposition 3 directly results from a recursive application of Lemma 1. Starting from the empty set, all the indices, sorted by decreasing order, are included successively.

Basically, it states that we can allow several discontinuities to be active at the same location i, but then, the next samples $i + 1, \ldots, i + n_i - 1$ must not host any discontinuity. This condition ensures that there are at most n_i discontinuities in the interval $\{i, \ldots, i + n_i - 1\}$ of length n_i. The adaptation of SBR consists in testing insertions into the current active set only if the above condition remains true.

2.6.4 Numerical simulations

Let us first consider the case $P = 1$, leading to the joint detection of discontinuities of order zero and one, *i.e.,* the piecewise affine approximation problem. We simulate noise-free data $\boldsymbol{y}^* = \boldsymbol{A}\boldsymbol{x}^*$ of size $m = 1000$ and with $\|\boldsymbol{x}^*\|_0 = 18$ discontinuities (see Fig. 2.3(a)). We use the result of Proposition 2 to compute the value λ_{max} below which the SBR output is not the empty set, and we run SBR with $\lambda_j = \lambda_{max} 10^{-j/2}$ for $j = 0, \ldots, J_{max}$, with $J_{max} = 20$. These λ-values provide a sequence of solutions at different sparsity levels. For comparison purpose, we also run 27 iterations of OMP and OLS.

The SBR approximation shown in Fig. 2.3(a) corresponds to the least λ-value $\lambda_{J_{max}}$. The recovery is exact. The "ℓ_2-ℓ_0" curves represented on Fig. 2.3(b) express the squared residual $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ versus the cardinality $\|\boldsymbol{x}\|_0$ for the output of each algorithm. Whatever the level of sparsity, SBR yields the least residual. We did the same experiment with noisy data $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{n}$, setting the SNR to 35 dB (see Figs. 2.3(c,d)). Again, the "ℓ_2-ℓ_0" curve corresponding to SBR lays below the OMP and OLS curves. For most sparsity levels, SBR outperforms the other algorithms. Note that for more noisy data (*e.g.,* SNR = 15 dB), the SBR and OLS curves coincide and still lay below the OMP curve.

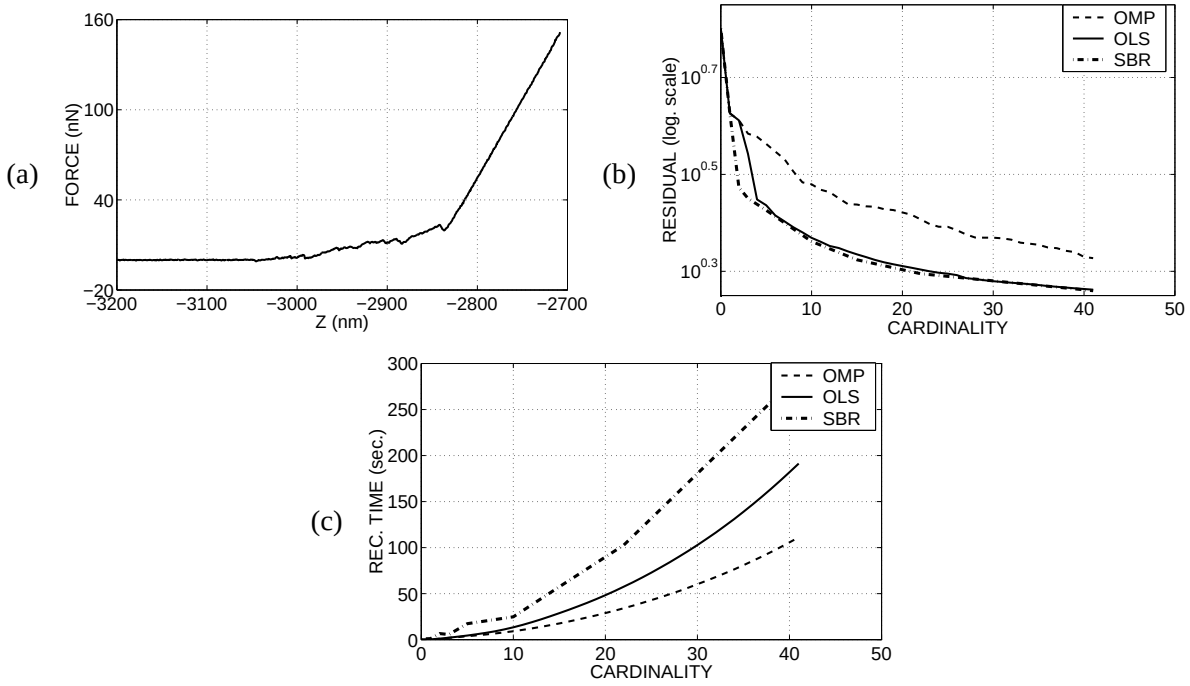
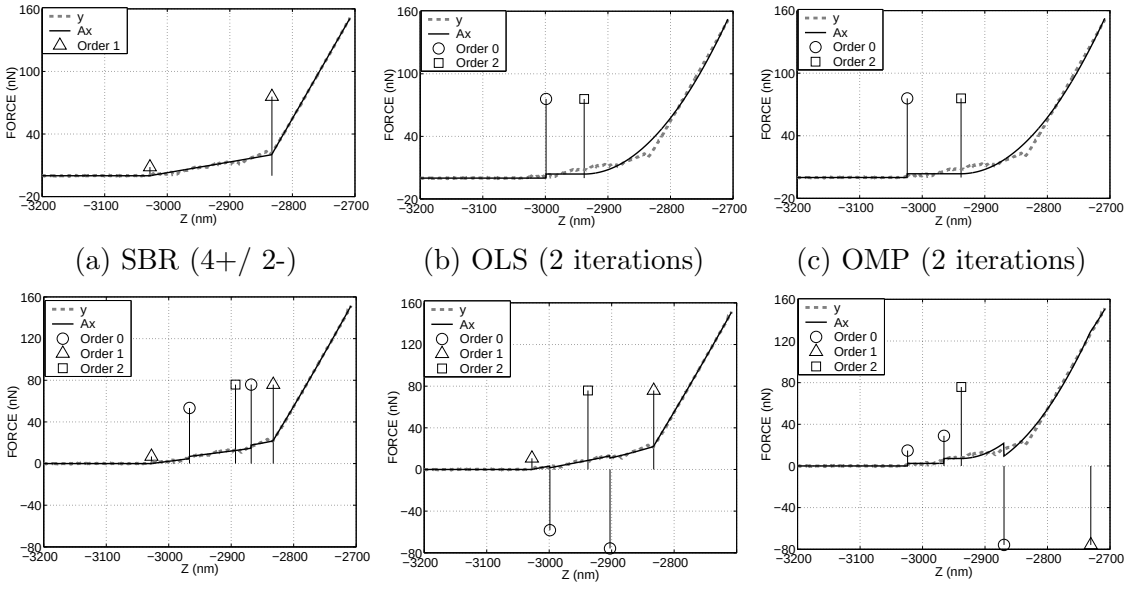


FIGURE 2.4 – AFM data processing : joint detection of discontinuities at orders 0, 1, and 2 (problem of size 2167×6498). (a) Experimental data showing the force evolution versus the probe-sample distance z. (b) Squared residual versus cardinality for the SBR, OLS, and OMP outputs. (c) Time of reconstruction versus cardinality for the three algorithms



(a) SBR (4+/ 2-) (b) OLS (2 iterations) (c) OMP (2 iterations)

(d) SBR (7+/ 2-) (e) OLS (5 iterations) (f) OMP (5 iterations)

FIGURE 2.5 – AFM data processing : joint detection of discontinuities at orders 0, 1, and 2. The estimated discontinuities $\boldsymbol{x}$ are represented with vertical spikes and with a label indicating the discontinuity order. The dashed gray and solid black curves represent the data signal $\boldsymbol{y}$ and its approximation $\boldsymbol{Ax}$, respectively. (a) SBR output of cardinality 2 : 4 insertions and 2 removals have been done (λ=2085). (b,c) OLS and OMP outputs after 2 iterations. (d,e,f) Same simulation with a lower λ-value (λ=66). The SBR output is of cardinality 5 (7 insertions and 2 removals) and we stop OLS and OMP after 5 iterations.

2.6.5 AFM data processing

In Atomic Force Microscopy (AFM), a force curve measures the interatomic forces exerting between a probe associated to a cantilever and a nano-object. More precisely, the recorded signal $z \mapsto y(z)$ shows the force evolution versus the probe-sample distance z, expressed in nanometers. Researching discontinuities (location, order, and amplitude) in a force curve is a challenging task because they are used to provide a precise characterization of the physico-chemical properties of the nano-object (topography, energy of adhesion, *etc.*) [2].

The data displayed on Fig. 2.4(a) are related to a bacterial cell *Shewanella putrefaciens* laying in aqueous solution, interacting with the tip of the AFM probe [52]. A force curve is recorded in two steps. Firstly, the tip is positioned far away from the sample. It is moved towards the sample until the contact is reached and the surface of the bacterial cell is deformed (approach curve). Secondly, the tip is retracted from the sample until it loses contact. The experimental curve shown on Fig. 2.4(a) is a retraction curve composed of $m = 2167$ force measurements. From right to left, three regions of interest can be distinguished. The linear region on the right characterizes the rigid contact between the probe and the sample. It describes the mechanical interactions of the cantilever and the sample. The rigid contact is maintained until $z \approx -2840$ nm. The interactions occurring in the interval $z \in [-3050, -2840]$ nm are adhesion forces during the retraction of the tip. In the flat part on the left, no interaction occurs as the cantilever has lost contact with the sample.

We search for the discontinuities of orders 0, 1, and 2. Similarly to the processing of simulated data, we run SBR with $J_{max} = 14$ λ-values and we run OLS and OMP until iteration 41. For each algorithm, we plot the "ℓ_2-ℓ_0" curve representing the squared residual $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ versus the cardinality $\|\boldsymbol{x}\|_0$, and a curve displaying the time of reconstruction versus the cardinality (see Figs. 2.4(b,c)). These figures show that the performance of SBR is at least equal and sometimes better than that of OLS. Both algorithms yield results that are far more accurate than OMP at the price of a larger computation time. However, notice that the recorded computation time always remains below 350 seconds in the case of SBR (in a Matlab implementation taking advantage of the block Toeplitz structure of the dictionary : see Section 2.4.4).

Fig. 2.5 displays the approximations yielded by the three algorithms for supports of cardinality 2 and 5. SBR actually runs during 6 iterations (4 insertions and 2 removals are performed) to reach a support of cardinality 2. This approximation is very accurate compared to the OMP and OLS results obtained after 2 iterations (Figs. 2.5(a,b,c)). SBR provides a very precise localization of both first order discontinuities, which are crucial information for the physical interpretation of the data. On the contrary, OLS does not succeed after two iterations ; it is able to locate accurately both discontinuities once 5 iterations have been performed (the desired discontinuities are the first and the last ones among the 5) while OMP fails even after 5 iterations (Figs. 2.5(d,e,f)). The residual yielded by the SBR approximation of cardinality 5 remains lower than the corresponding OLS and OMP residuals.

In order to better understand the forward and backward moves (respectively, insertions and removals) occurring during the SBR iterations, we display in Tab. 2.4 the residual $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ and the cardinality of each iterate for both SBR executions. Because SBR is a descent algorithm, the penalized cost $\mathcal{J}(\boldsymbol{x}; \lambda)$ keeps decreasing but when a removal occurs, $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$ increases. For the coarse approximation of Fig. 2.5(a), the residual is much smaller than the residual yielded by OLS since two configurations of cardinality 2 have been explored (see the left table and the rows in bold). We observed the same behavior for the finer approximation of cardinality 5 (right table).

2.6.6 Discussion

In the two previous subsections, we chose to compare SBR with OMP and OLS. We did not consider simpler algorithms like MP which are well suited to rapidly solve easier problems in which

the dictionary columns are almost orthogonal. Because SBR involves more complex operations (matrix inversions), we chose to compare it with OMP and OLS which also require to solve at least one least-square problem per iteration. Their target is to provide results which are more accurate than the MP approximations in the case of difficult problems.

Up to our knowledge, the only minimization algorithm dedicated to the ℓ_0-penalized cost function $\mathcal{J}(\boldsymbol{x};\lambda) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{x}\|_0$ is Blumensath and Davies' Iterative Hard Thresholding (IHT) [33]. It relies on gradient based iterations of the form $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{A}^t(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x})$, followed by the thresholding to 0 of all the non-zero components x_i such that $|x_i| \leqslant \lambda^{0.5}$. We tested this version of IHT on both deconvolution and discontinuity detection problems and we observed that it is less efficient than the standard version of IHT, dedicated to the ℓ_0-constrained problem. In the constrained version, the k components $|x_i|$ having the largest amplitudes are kept and the others are being thresholded. Generally speaking, we observed that IHT is competitive when the correlation between any pair of dictionary columns is limited, but for highly correlated dictionaries, IHT needs a very large number of iterations ($\mathcal{O}(m^2)$) to reach convergence. SBR seems to be better suited to such difficult problems. It is less sensitive to the initial solution and "skips" some local minimizers having a large cost $\mathcal{J}(\boldsymbol{x};\lambda)$. We here recall that according to Proposition 1, each SBR iterate is almost surely a local minimizer of $\mathcal{J}(\boldsymbol{x};\lambda)$.

In order to link up our approach to the forward-backward algorithm of [36], we also tested an OMP-like adaptation of SBR in which only one least-square problem is solved per iteration, instead of n. This adaptation consists in replacing the selection rule (2.11) in the following way. When an insertion $\mathcal{Q} \cup \{i\}$ is tested, all the active components x_j are kept constant and x_i is set to the minimizer of $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_{\mathcal{Q}} - x_i\boldsymbol{a}_i\|^2$. This leads to an approximation of $\mathcal{K}_{\mathcal{Q}\bullet i}(\lambda)$ without solving any least-square problem. Similarly, the removal test consists in setting x_i to 0 and leaving the other components x_j unchanged. In brief, this adapted version is an algorithm aimed at the minimization of $\mathcal{J}(\boldsymbol{x};\lambda)$ at a cost which is comparable to that of OMP. In all our trials, SBR performs better than the OMP-like version except in very simple cases (limited correlation between the columns $\boldsymbol{a}_i$) where both versions yield the same result. The performance of the OMP-like version fluctuates below or above that of OMP but is almost always far less accurate than the OLS and SBR approximations.

2.7 Conclusion

We have evaluated the SBR algorithm on two problems in which the dictionary columns are highly correlated. SBR provides solutions which are at least as accurate as the OLS solutions and sometimes more accurate, with a cost of the same order of magnitude. For these difficult problems, MP and OMP provide poor approximations within a lower computation time. Compared to OLS, we believe that performing removals is the price to pay if one expects an enhanced quality approximation. Although removals rarely occur in comparison with the insertions, they play an important role in the enhancement of the approximation.

In the proposed approach, the main difficulty relies in the choice of the λ-value. If a specific sparsity level or approximation residual is desired, one can resort to a trial and error procedure in which a number of λ-values are tried until the desired approximation level is found. In [21], we sketched a continuation version in which a series of SBR solutions are successively estimated with a decreasing level of sparsity λ, and the λ-values are recursively computed. The first λ-value is set to $\lambda_0 = +\infty$, and at a given value λ_i, the initial solution (input of SBR) is set to the SBR output at $\lambda = \lambda_{i-1}$. This continuation version provides promising results and will be the subject of a future extended contribution. A similar perspective is actually proposed by Zhang to generalize his FoBa algorithm in a path-following algorithm (see the discussion section in [36]).

Another important perspective is to investigate whether SBR can guarantee exact recovery in the noise-free case under some conditions on matrix $\boldsymbol{A}$ and on the unknown sparse signal $\boldsymbol{x}^*$.

In the simulations done in Section 2.5, we observed that SBR is able to exactly recover two close Gaussian features whatever their distance, provided that the hyperparameter λ is sufficiently small. This promising result is a first step towards a more general theoretical study. The FoBa algorithm [36] yields exact recovery results for problems satisfying the Restricted Isometry Property (RIP). Since the structure of SBR is somewhat close to that of FoBa, we expect that SBR shares similar theoretical properties. We will investigate whether the proofs provided in [36] are extendable to SBR.

Chapitre 3

A continuation algorithm for ℓ_0-penalized least-square optimization

3.1 Introduction

In Chapter 2, sparse signal approximation is formulated as the ℓ_0-penalized least-square optimization, *i.e.,* the minimization of a cost function of the form $\mathcal{J}(\boldsymbol{x};\lambda) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{x}\|_0$. We proposed the Single Best Replacement algorithm (SBR), which is a forward-backward descent algorithm in which the set of active columns of $\boldsymbol{A}$ is modified by one element (insertion or removal of a dictionary column). It is of great interest to design an extended algorithm providing a sequence of approximations with various sparsity levels (SBR solutions for various λ-values or for various k-values, where k is the cardinality of the SBR output). Running SBR for a large number of λ-values is time consuming and inefficient if the λ-values are not chosen in an appropriate manner : we observed that many λ-values (close or not) lead to the same SBR output. The algorithm proposed in this chapter aims at minimizing $\mathcal{J}(.\,;\lambda)$ for many characteristic λ-values, yielding a sequence of coarse to fine approximations. It is based on recursive calls to SBR with a set of decreasing **λ-values that are adaptive to the data**. The algorithm is actually inspired by the existing efficient continuation algorithms (named LARS and homotopy) dedicated to the least-square optimization under the ℓ_1-norm constraint $\|\boldsymbol{x}\|_1 \leqslant t$ [8]. These algorithms are based on the piecewise affinity of the solution path of such problems : the optimal solution $\boldsymbol{x}(t)$ is a piecewise affine function of t. When using the ℓ_0-norm, the solution path is piecewise constant. Although we design an algorithm which estimates a piecewise constant solution path, we cannot ensure that the estimated path is the "optimal path". This is a major difference with the algorithms dedicated to mixed ℓ_2-ℓ_1 problems.

The chapter is organized as follows. In Section 3.2, we recall the main notations and formulations of the mixed ℓ_2-ℓ_0 optimization problems (ℓ_0-constrained and ℓ_0-penalized least squares). We introduce the notion of solution path by considering all λ-values (all sparsity levels). In Section 3.3, we propose the so-called Continuation SBR algorithm (CSBR) based on recursive calls to SBR with appropriate λ-values and the last SBR output as initial solution. In Section 3.4, the CSBR algorithm is applied to two inverse problems : a sparse spike train deconvolution problem and the discontinuity detection problem already introduced in Chapters 1 and 2. CSBR is compared with two other algorithms : ℓ_1 regression and Iterative Hard Thresholding. We also propose a simulated problem in which it is possible to tune the mutual coherence of the dictionary $\boldsymbol{A}$. We show the effectiveness of CSBR, in particular when the mutual coherence is large, *i.e.,* when the columns of $\boldsymbol{A}$ are highly correlated.

3.2 Mixed ℓ_2-ℓ_0 optimization

3.2.1 Definitions and working assumptions

We denote by $m \times n$ the size of matrix $\boldsymbol{A}$ (m can be either lower or greater than n). The observation signal $\boldsymbol{y}$ and the weight vector $\boldsymbol{x}$ are thus of size $m \times 1$ and $n \times 1$, respectively. For $k \leqslant n$, the ℓ_0-constrained least-square problem reads :

$$\min_{\|\boldsymbol{x}\|_0 \leqslant k} \{\mathcal{E}(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2\}. \tag{3.1}$$

For a given $\lambda > 0$, we define the set of minimizers of the ℓ_0-penalized least-square problem :

$$\mathcal{X}(\lambda) = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\arg\min} \{\mathcal{J}(\boldsymbol{x}; \lambda) = \mathcal{E}(\boldsymbol{x}) + \lambda \|\boldsymbol{x}\|_0\}. \tag{3.2}$$

By extension, we define $\mathcal{X}(+\infty) = \{\boldsymbol{0}\}$.

As in Chapter 2, we assume that the unique representation property (URP) holds, *i.e.,* that any $\min(m, n)$ columns of $\boldsymbol{A}$ are linearly independent. In practice, this assumption can be relaxed provided that there exists a simple way to check that a subset of columns are independent from each other (see Section VI-C of Chapter 2 for details).

Definition 5. *We denote by $\mathcal{Q} \subseteq \{1, \ldots, n\}$ the active set, i.e., the set of selected columns (with* $\text{Card}\,[\mathcal{Q}] \leqslant \min(m, n)$*). Let $\boldsymbol{x}_\mathcal{Q}$ denote the associated least-square solution :*

$$\boldsymbol{x}_\mathcal{Q} \triangleq \underset{\mathcal{S}(\boldsymbol{x}) \subseteq \mathcal{Q}}{\arg\min}\, \mathcal{E}(\boldsymbol{x}) \tag{3.3}$$

where $\mathcal{S}(\boldsymbol{x}) \subseteq \{1, \ldots, n\}$ stands for the support of $\boldsymbol{x}$. Let $\mathcal{E}_\mathcal{Q} \triangleq \mathcal{E}(\boldsymbol{x}_\mathcal{Q})$ and

$$\mathcal{J}_\mathcal{Q}(\lambda) \triangleq \mathcal{J}(\boldsymbol{x}_\mathcal{Q}; \lambda) = \mathcal{E}_\mathcal{Q} + \lambda \|\boldsymbol{x}_\mathcal{Q}\|_0. \tag{3.4}$$

3.2.2 Solution path

Definition 6. *The solution path is defined as the union of sets $\mathcal{X} = \bigcup_{\{\lambda > 0\}} \mathcal{X}(\lambda)$. It is characterized by the function $\lambda \mapsto \mathcal{J}(\lambda) \triangleq \min_{\boldsymbol{x} \in \mathbb{R}^n} \mathcal{J}(\boldsymbol{x}; \lambda) = \min_{\mathcal{Q} \subseteq \mathbb{R}^n} \mathcal{J}_\mathcal{Q}(\lambda)$.*

Definition 6 is illustrated on Fig. 3.1. Geometrically, each affine function $\lambda \mapsto \mathcal{J}_\mathcal{Q}(\lambda) = \mathcal{E}_\mathcal{Q} + \lambda \|\boldsymbol{x}_\mathcal{Q}\|_0$ yields a 2D line. The curve representing the minimal cost function $\mathcal{J}(\lambda)$ (plain lines) is the concave envelope of the set of lines $\mathcal{J}_\mathcal{Q}(\lambda)$ for all possible supports $\mathcal{Q}$. On the example of Fig. 3.1, the solution path is formed of the supports $\mathcal{Q}_0$, $\mathcal{Q}_1$, and $\mathcal{Q}_2$ (and the possible other sets yielding the same three lines).

We now develop the proposed CSBR algorithm which aims at providing a sequence of approximations for decreasing λ-values, computed adaptively to the data. Notice that the proposed algorithm is sub-optimal : it is not guaranteed to find the elements of $\mathcal{X}(\lambda)$.

3.3 Continuation algorithm

In subsection 3.3.1, we briefly recall the SBR algorithm whose goal is to minimize $\mathcal{Q} \mapsto \mathcal{J}_\mathcal{Q}(\lambda)$ for a given λ-value. Our strategy is based on recursive calls to SBR with decreasing λ-values that are adaptive to the data and recursively computed. In subsection 3.3.2, we show that it is possible

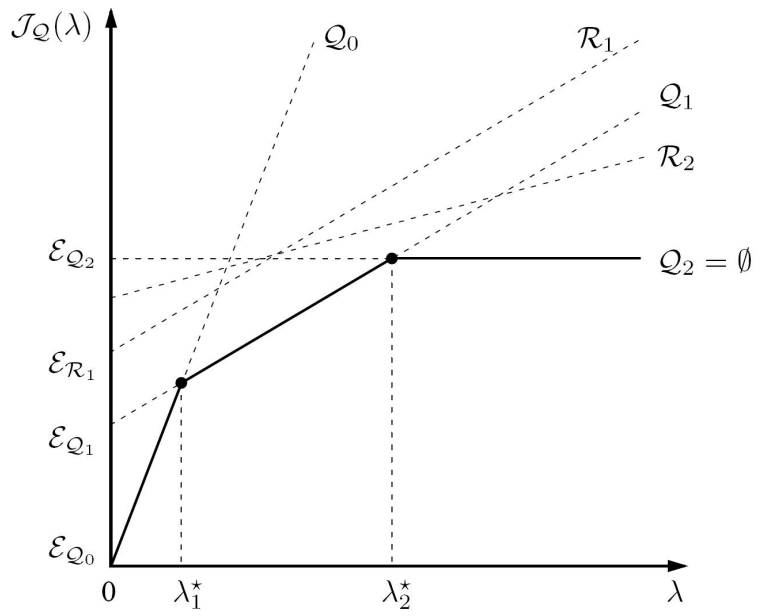


FIGURE 3.1 – Representation of a few lines $\lambda \mapsto \mathcal{J}_{\mathcal{Q}}(\lambda) = \mathcal{E}_{\mathcal{Q}} + \lambda \|\boldsymbol{x}_{\mathcal{Q}}\|_0$ for different supports $\mathcal{Q}$. Note that two lines $\mathcal{J}_{\mathcal{Q}}$ and $\mathcal{J}_{\mathcal{Q}'}$ may coincide. The optimal cost function $\lambda \mapsto \mathcal{J}(\lambda)$ is represented in plain line. It can be interpreted as the "concave envelope" of all lines $\lambda \mapsto \mathcal{J}_{\mathcal{Q}}(\lambda)$ (for all possible supports $\mathcal{Q}$). Here, the solution path is described by supports $\mathcal{Q}_0$ (and the possible other sets yielding the same line) for $\lambda < \lambda_1^\star$, $\mathcal{Q}_1$ (of lower cardinality) in $(\lambda_1^\star, \lambda_2^\star)$, and $\mathcal{Q}_2 = \emptyset$ for $\lambda > \lambda_2^\star$. The supports $\mathcal{R}_1$ and $\mathcal{R}_2$ are such that $\forall \lambda$, $\mathcal{J}_{\mathcal{R}}(\lambda) > \mathcal{J}(\lambda)$.

to compute the characteristic λ-values recursively in a simple closed form expression. Finally, we present the continuation algorithm (CSBR) in subsection 3.3.3.

Remark 2. *In Chapter 2, we have shown that provided that the data $\boldsymbol{y}$ are corrupted with "non degenerate" noise, the SBR iterates satisfy $\|\boldsymbol{x}_{\mathcal{Q}}\|_0 = \mathrm{Card}\,[\mathcal{Q}]$ almost surely. In the SBR algorithm, we thus chose to minimize*

$$\mathcal{K}_{\mathcal{Q}}(\lambda) \triangleq \mathcal{E}_{\mathcal{Q}} + \lambda \mathrm{Card}\,[\mathcal{Q}] \tag{3.5}$$

which can be more easily computed than $\mathcal{J}_{\mathcal{Q}}(\lambda)$. In the following, we still consider the minimization of $\mathcal{K}_{\mathcal{Q}}(\lambda)$ instead of $\mathcal{J}_{\mathcal{Q}}(\lambda)$.

3.3.1 Single Best Replacement

We briefly recall the principle of a single SBR iteration, and then detail the input-output relationship and the storage of useful information, namely the best SBR iterates.

Principle of a SBR iteration

Let us denote by $\mathcal{Q} \bullet i$ a single replacement, *i.e.,* the insertion or removal of a column index i into/from the current active set $\mathcal{Q}$:

$$\mathcal{Q} \bullet i \triangleq \begin{cases} \mathcal{Q} \cup \{i\} & \text{if } i \notin \mathcal{Q}, \\ \mathcal{Q} \backslash \{i\} & \text{otherwise.} \end{cases} \tag{3.6}$$

TABLE 3.1 – SBR algorithm. The line in bold is an extra storage which is useful to avoid re-computations while running CSBR.

Inputs : $\boldsymbol{A}$, $\boldsymbol{y}$, λ, and initial active set $\mathcal{Q}^{\text{init}}$ (Card $\left[\mathcal{Q}^{\text{init}}\right] \leqslant \min(m,n)$)
Step 1 : Set $\mathcal{Q} = \mathcal{Q}^{\text{init}}$.
Step 2 : For $i \in \{1,\dots,n\}$, compute $\mathcal{K}_{\mathcal{Q}\bullet i}(\lambda)$.
Store the two maximal values $\delta\mathcal{E}_{i_1}$ and $\delta\mathcal{E}_{i_2}$ of $\mathcal{E}_{\mathcal{Q}} - \mathcal{E}_{\mathcal{Q}\cup\{i\}}$ for $i \notin \mathcal{Q}$.
Compute ℓ using (3.7).
If $\mathcal{K}_{\mathcal{Q}\bullet\ell}(\lambda) < \mathcal{K}_{\mathcal{Q}}(\lambda)$,
Set $\mathcal{Q} = \mathcal{Q} \bullet \ell$.
else,
Terminate SBR.
End if.
Go to Step 2.
Outputs : Active set $\mathcal{Q} = \text{SBR}(\mathcal{Q}^{\text{init}}; \lambda)$.
Two maximal values $\delta\mathcal{E}_{i_1}$ and $\delta\mathcal{E}_{i_2}$ of $\delta\mathcal{E}_i = \mathcal{E}_{\mathcal{Q}} - \mathcal{E}_{\mathcal{Q}\cup\{i\}}$.

An SBR iteration is based on the computation of $\mathcal{K}_{\mathcal{Q}\bullet i}(\lambda)$ for all i (n insertion and removal trials) and the selection of the replacement for which $\mathcal{K}_{\mathcal{Q}\bullet i}(\lambda)$ is minimal :

$$\ell \in \underset{i\in\{1,\dots,n\}}{\arg\min}\ \mathcal{K}_{\mathcal{Q}\bullet i}(\lambda). \tag{3.7}$$

Termination, initial active set, and outputs

The SBR algorithm is summarized in Tab. 3.1. It is important to notice that no stopping condition is necessary : SBR terminates when no replacement decreases the cost function. It terminates after a finite number of iterations because there are a finite number of possibilities for the active set $\mathcal{Q}$.

In Chapter 2, we set the initial active set to the empty support (no column is selected), the input-output relation reading

$$\mathcal{Q}(\lambda) = \text{SBR}(\emptyset; \lambda).$$

In the following, we perform recursive calls to SBR at decreasing λ-values with the last SBR output as initial solution. We will use the following notation for the j-th call to SBR :

$$\mathcal{Q}_j = \text{SBR}(\mathcal{Q}_{j-1}; \lambda_j). \tag{3.8}$$

When calling SBR recursively, it is also helpful to store and update the "best SBR iterates" in a table $\mathcal{Q}_k^{\text{best}}$. $\mathcal{Q}_k^{\text{best}}$ is defined as the SBR iterate of cardinality k having the least squared error $\mathcal{E}_k^{\text{best}}$, considering all iterates of cardinality k reached during all calls to SBR. The tables $\mathcal{Q}_k^{\text{best}}$ and $\mathcal{E}_k^{\text{best}}$ are being updated whenever SBR is called (at each iteration (3.8)).

3.3.2 Recursive computation of λ

Let us consider the execution of SBR for a given λ-value, yielding the support $\mathcal{Q} = \text{SBR}(\mathcal{Q}_{j-1}; \lambda)$ as output [1]. Obviously, the output of $\text{SBR}(\mathcal{Q}; \lambda)$ is also equal to $\mathcal{Q}$ since the stopping condition of

1. For readability reasons, the notations are simplified in this paragraph. In the continuation algorithm, λ and the SBR output $\mathcal{Q}$ will be replaced by λ_j and $\mathcal{Q}_j$, according to (3.8).

SBR is fulfilled for support $\mathcal{Q}$. The recursive computation of λ is based on the fact that the output of $\mathrm{SBR}(\mathcal{Q};\mu)$ is equal to $\mathcal{Q}$ when μ belongs to an interval surrounding λ. We now show that it is possible to exactly determine this interval, and especially its lower bound $\tilde{\lambda}$, referred to as a *critical value*.

Definition 7. *Let us distinguish the two cases where the termination of SBR is strict and large.*

- *If* $\forall i$, $\mathcal{K}_{\mathcal{Q}}(\lambda) < \mathcal{K}_{\mathcal{Q}\bullet i}(\lambda)$, *we define the next critical value as the lower bound* $\tilde{\lambda} < \lambda$ *such as for all* $\mu \in (\tilde{\lambda}, \lambda]$, *the output of* $\mathrm{SBR}(\mathcal{Q};\mu)$ *is equal to* $\mathcal{Q}$, *i.e.,* $\mathrm{SBR}(\mathcal{Q};\mu)$ *terminates in its first iteration.*
- *If* $\mathcal{K}_{\mathcal{Q}}(\lambda) = \mathcal{K}_{\mathcal{Q}\bullet i}(\lambda)$ *holds for at least one value of* i, *we set* $\tilde{\lambda} = \lambda$.

Computation of the next critical value $\tilde{\lambda}$

By definition of $\tilde{\lambda}$, for $\mu \in (\tilde{\lambda}, \lambda]$, $\mathrm{SBR}(\mathcal{Q};\mu)$ stops in its first iteration, yielding $\mathcal{Q}$ as output. The termination condition $\forall i$, $\mathcal{K}_{\mathcal{Q}}(\mu) \leqslant \mathcal{K}_{\mathcal{Q}\bullet i}(\mu)$ rereads :

$$\forall i,\ \mathcal{E}_{\mathcal{Q}} - \mathcal{E}_{\mathcal{Q}\bullet i} \ \leqslant\ \mu\big(\mathrm{Card}\,[\mathcal{Q}\bullet i] - \mathrm{Card}\,[\mathcal{Q}]\big). \tag{3.9}$$

For $\mu < \tilde{\lambda}$, (3.9) does not hold anymore.

Proposition 4. *The next critical value* $\tilde{\lambda}$ *is equal to :*

$$\tilde{\lambda} = \delta\mathcal{E}_{i_1} \triangleq \max_{i\notin\mathcal{Q}}\big(\delta\mathcal{E}_i \triangleq \mathcal{E}_{\mathcal{Q}} - \mathcal{E}_{\mathcal{Q}\cup\{i\}}\big), \tag{3.10}$$

where i_1 *denotes an index* i *for which* $\delta\mathcal{E}_i$ *is maximal.*

Proof. When $\mu < \lambda$, let us consider the inequalities (3.9) in the insertion and removal cases.

- When $\bullet = \backslash$, both terms on the left- and right-hand sides of the inequalities are strictly negative since $\mathrm{Card}\,[\mathcal{Q}\backslash\{i\}] = \mathrm{Card}\,[\mathcal{Q}] - 1$ and (3.9) holds for $\mu = \lambda$. Therefore, the inequalities still holds for all $\mu < \lambda$.
- When $\bullet = \cup$, both terms on the left- and right-hand sides of the inequalities are non-negative since $\mathcal{E}_{\mathcal{Q}} - \mathcal{E}_{\mathcal{Q}\cup\{i\}} \geqslant 0$ and (3.9) holds for $\mu = \lambda$. Because $\mathrm{Card}\,[\mathcal{Q}\cup\{i\}] = \mathrm{Card}\,[\mathcal{Q}] + 1$, (3.9) remains valid for $\mu < \lambda$ *if and only if* $\forall i$, $\mathcal{E}_{\mathcal{Q}} - \mathcal{E}_{\mathcal{Q}\cup\{i\}} \leqslant \mu$.

Finally, (3.9) remains valid for $\mu < \lambda$ *if and only if* $\mu \geqslant \max_{i\notin\mathcal{Q}} \delta\mathcal{E}_i$. □

Computation of the next λ-value

Proposition 4 states that the output of $\mathrm{SBR}(\mathcal{Q};\mu)$ is changing at $\mu = \tilde{\lambda}$: when $\mu > \tilde{\lambda}$, SBR stops in its first iteration while when $\mu < \tilde{\lambda}$ and μ is close enough to $\tilde{\lambda}$, SBR performs at least one iteration (the insertion of any index $i_1 \notin \mathcal{Q}$ for which $\delta\mathcal{E}_{i_1}$ is maximal).

Since the behavior of SBR is changing at $\tilde{\lambda}$, we do not call $\mathrm{SBR}(\mathcal{Q};\mu)$ with $\mu = \tilde{\lambda}$. We rather consider the open interval $(\delta\mathcal{E}_{i_2}, \delta\mathcal{E}_{i_1})$ formed by the two largest values of $\delta\mathcal{E}_i$ ($i \notin \mathcal{Q}$). It is easy to check that if $\mu \in (\delta\mathcal{E}_{i_2}, \delta\mathcal{E}_{i_1})$, the first iteration of $\mathrm{SBR}(\mathcal{Q};\mu)$ always leads to the inclusion of an index $i_1 \notin \mathcal{Q}$ for which $\delta\mathcal{E}_{i_1}$ is maximal. We finally choose to call $\mathrm{SBR}(\mathcal{Q};\lambda_{\mathrm{next}})$ with

$$\lambda_{\mathrm{next}} = (\delta\mathcal{E}_{i_1} + \delta\mathcal{E}_{i_2})/2. \tag{3.11}$$

To compute λ_{next}, we need to sort the values of $\delta\mathcal{E}_i$ for all indices $i \notin \mathcal{Q}$. Their computation is mentioned in the SBR algorithm given in Tab. 3.1.

TABLE 3.2 – CSBR algorithm. During each call to SBR, the tables $\mathcal{Q}_k^{\text{best}}$ and $\mathcal{E}_k^{\text{best}}$ which store the best SBR iterates and their related squared errors are being updated.

Inputs : $\boldsymbol{A}$, $\boldsymbol{y}$, $K_{\max}$ and/or $\mathcal{E}_{\text{MIN}}$ (residual threshold)
Step 1 : Initialization : $\lambda_0 = +\infty$, $\mathcal{Q}_0 = \emptyset$. Set $j = 1$. Compute $\tilde{\lambda}_1$ and λ_1 using (3.10) and (3.11).
Step 2 : $[\mathcal{Q}_j, \delta\mathcal{E}_{i_1}, \delta\mathcal{E}_{i_2}] = \text{SBR}(\mathcal{Q}_{j-1}; \lambda_j)$. Compute $\tilde{\lambda}_{j+1} = \delta\mathcal{E}_{i_1}$ and $\lambda_{j+1} = (\delta\mathcal{E}_{i_1} + \delta\mathcal{E}_{i_2})/2$. If $(\text{Card}\,[\mathcal{Q}_j] \geqslant K_{\max})$ or $(\mathcal{E}_{\mathcal{Q}_j} \leqslant \mathcal{E}_{\text{MIN}})$, Terminate CSBR. End if Do $j = j + 1$ and go to Step 2.
Outputs : Best SBR iterates $\mathcal{Q}_k^{\text{best}}$ and their residuals $\mathcal{E}_k^{\text{best}}$ $(k = 0, \ldots, K_{\max})$. Critical and actual values $\tilde{\lambda}_j$ and λ_j (for all j).

3.3.3 CSBR algorithm

We introduce a first version of CSBR consisting of alternating calls to SBR for decreasing λ-values and the estimation of the next (lower) λ-value at which SBR is called. The final version of CSBR is a straightforward adaptation enabling the estimation of a sequence of k-sparse approximations for all consecutive values of $k = 0, 1, \ldots$

First version

We denote by λ_j and $\mathcal{Q}_j$ the λ and $\mathcal{Q}$-values at iteration j, with $\mathcal{Q}_j = \text{SBR}(\mathcal{Q}_{j-1}; \lambda_j)$. The critical value $\tilde{\lambda}_{j+1} < \lambda_j$ and the next value $\lambda_{j+1} < \tilde{\lambda}_{j+1}$ for which SBR is called are computed using (3.10) and (3.11).

Each algorithm output $\boldsymbol{x}_{\mathcal{Q}_j}$ can be seen as a sub-optimal solution to the minimization of $\mathcal{J}(.\,; \lambda)$ *for all* $\lambda \in (\tilde{\lambda}_{j+1}, \tilde{\lambda}_j)$. In other words, CSBR estimates a continuum of solutions which is piecewise constant with respect to λ. The $\tilde{\lambda}_j$ values form the "estimated solution path". The structure of CSBR is summarized in Tab. 3.2. Initially, λ_0 is set to $+\infty$, yielding $\mathcal{Q}_0 = \emptyset$. We set two stopping conditions relying on the CSBR iterates $\mathcal{Q}_j$:

1. CSBR is stopped when a CSBR iterate is of cardinality $\text{Card}\,[\mathcal{Q}_j] \geqslant K_{\max}$;
2. CSBR is stopped when a CSBR iterate yields a squared error $\mathcal{E}_{\mathcal{Q}_j}$ that is lower than $\mathcal{E}_{\text{MIN}}$.

The user can activate either of them or both.

SBR is never stopped before convergence. If the first stopping rule holds, the last SBR output is of cardinality $K_{\max}$ or more. The stopping condition $\mathcal{E}_{\mathcal{Q}_j} \leqslant \mathcal{E}_{\text{MIN}}$ is especially useful when dealing with real data. When the desired cardinality to reach is unknown *a priori* and when the estimation of the noise variance is possible, we suggest to set $\mathcal{E}_{\text{MIN}}$ based on the knowledge of the noise variance.

Final version

CSBR can be easily adapted to provide sparse approximations of cardinality k for consecutive values $k = 0, 1, \ldots$ up to a storage of the best intermediate SBR iterates (see 3.3.1). Once CSBR terminates, this sequence is formed of a support (SBR iterate) of cardinality k for all $k \in \{0, \ldots, K\}$ where K is the maximum cardinality reached.

3.4 Numerical simulations and real data processing

We study the performance of the CSBR algorithm by applying it to three inverse problems. The first problem is the joint detection of discontinuities in a signal, which is already introduced in Chapters 1 and 2, allows us to process real AFM data. The second problem is the deconvolution of a sparse spike train with a known impulse response. The third problem is a simulated problem in which it is possible to tune the mutual coherence of the dictionary $\boldsymbol{A}$. We study the effectiveness of CSBR depending on the mutual coherence of the dictionary. Here we define the mutual coherence of matrix $\boldsymbol{A}$ as [30] :

$$\mu(\boldsymbol{A}) = \max_{i \neq j} \frac{< \boldsymbol{a}_i, \boldsymbol{a}_j >}{\|\boldsymbol{a}_i\| \|\boldsymbol{a}_j\|} \tag{3.12}$$

where $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ is the i-th and j-th column of $\boldsymbol{A}$ respectively.

In Chapter 2, we compared SBR with Orthogonal Matching Pursuit (OMP) and Orthogonal Least Squares (OLS) which are forward deterministic algorithms whose numerical costs are cheaper than that of SBR. We showed that SBR outperforms the two other algorithms in cases where some of the dictionary columns are highly correlated, *i.e.,* the mutual coherence is high. We developed a similar comparative study in the conference paper [21] showing that CSBR outperforms OMP and OLS at the cost of a larger computation time (but of the same order of magnitude as OLS). Here, we choose to compare CSBR with two other algorithms which are very popular in the sparse approximation literature : the homotopy algorithm [8] and the Iterative Hard Thresholding (IHT) algorithm [33]. Homotopy algorithm, which was developed for ℓ_1-regression, has been already introduced in the Introduction section, so here we just introduce IHT.

Iterative Hard Thresholding was developed to tackle ℓ_0-constrained least-squares problem. The basic iteration is :

$$\boldsymbol{x}^j = \phi_k^h(\boldsymbol{x}^{j-1} + \boldsymbol{A}^T(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{j-1})) \tag{3.13}$$

where $\phi_k^h(\bullet)$ is the hard thresholding function keeping only the k largest entries, and $\boldsymbol{x}^j$ the j-th iterate. In the simulation, we use the Matlab function *hard_l0_Mterm* which is downloaded from the author's personal page. For ℓ_1-regression, we use the Matlab function *SolveLasso_Flops* coming from the software package *SparseLab*.

3.4.1 Discontinuity detection

Problem formulation

In Chapter 2, we formulated the joint discontinuity detection at different orders in a signal $\boldsymbol{y}$ as ℓ_0-constrained least-square problem (3.1). The piecewise polynomial dictionary $\boldsymbol{A}$ is constructed such that each column consists only one discontinuity at unique location and unique order. The discontinuity detection in $\boldsymbol{y}$ becomes the subset selection of columns in matrix $\boldsymbol{A}$. *i.e.,* the activation of a column $\boldsymbol{a}_i$, indicates the detection of a discontinuity in signal $\boldsymbol{y}$ at the same location with the same order as that in $\boldsymbol{a}_i$.

Scalar and vectorial subset selection

As mentioned in Chapter 1, there are two alternative strategies for discontinuity detection : the scalar subset selection and the vectorial subset selection.

The scalar subset selection consists in the direct use of CSBR with the block Toeplitz dictionary $\boldsymbol{A}$. In this version, if a discontinuity of order p is activated at location i, the discontinuities of order $q \neq p$ are not constrained to be activated at the same location i. To perform vectorial subset selection, one must adapt the SBR algorithm in such a way that an insertion test involves the joint

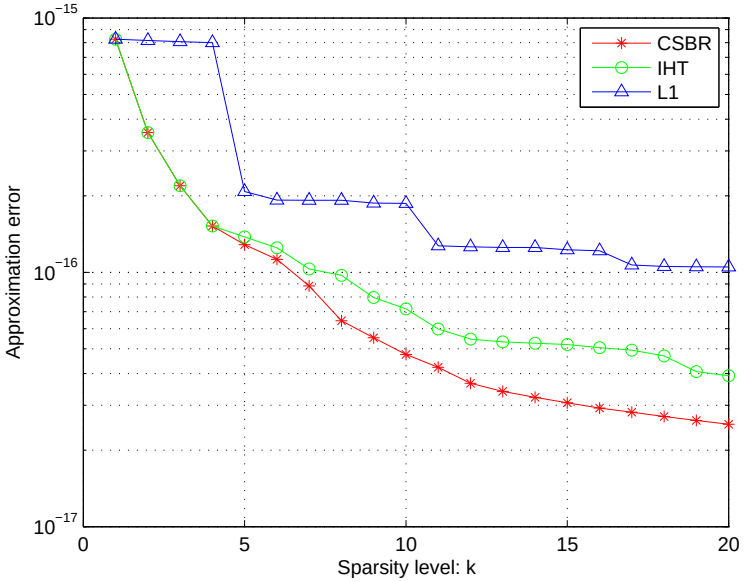
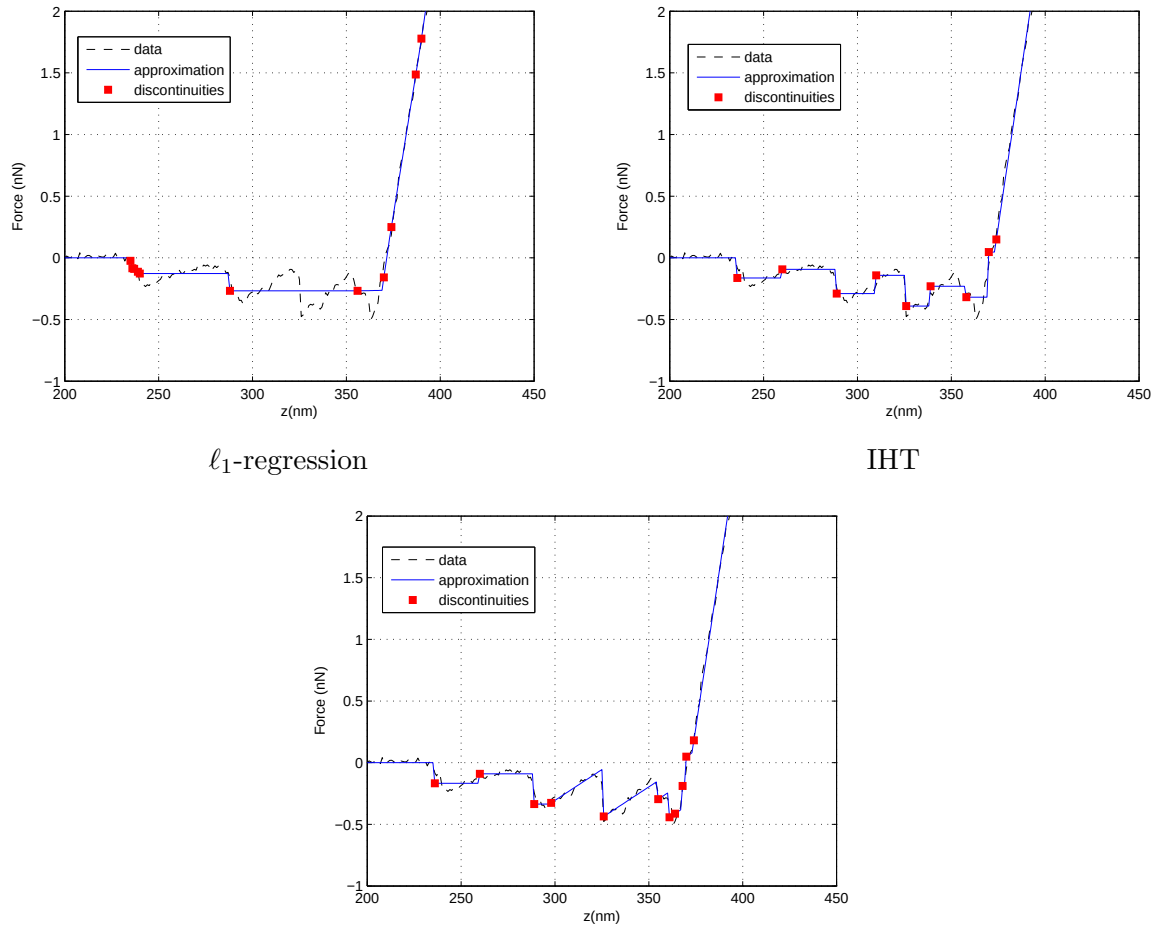


FIGURE 3.2 – Approximation performance of CSBR, IHT and ℓ_1-regression on force curve processing.



ℓ_1-regression IHT

CSBR

FIGURE 3.3 – Reconstructed force curves and detected discontinuities. Because of zoom in, some discontinuities are not in the visible region.

activation of the discontinuities for all orders $p = 0, 1, \ldots, P-1$ at a given location (insertion of P new columns into the active set of columns) while a removal test consists in removing the P columns associated with a given active location (the discontinuities at all orders $p = 0, \ldots, P-1$ are inactivated). We refer the reader to [26] for more details on the vectorial extension of SBR and CSBR, and to [53] and for the brief description of an algorithm which is a vectorial extension of OLS. This algorithm is being applied to the experimental force curves in Chapter 4.

Application of CSBR to an experimental force curve

In this experiment, we consider the joint detection of discontinuities at different orders in real AFM data $\boldsymbol{y}$ of size 516×1. This problem was already formulated in Chapter 2 as ℓ_0-constrained least-square problem, so here we just recall the definition of dictionary $\boldsymbol{A}$. Firstly, the pth order piecewise polynomial subdictionary is defined as :

$$\boldsymbol{A}_p = \begin{pmatrix} 1^p & 0 & \cdots & 0 \\ 2^p & 1^p & & \vdots \\ 3^p & 2^p & & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \vdots & \vdots & \cdots & 1^p \\ m^p & (m-1)^p & \cdots & 2^p \end{pmatrix} \tag{3.14}$$

$m = 516$ is force curve length. The dictionary for the joint detection of discontinuities at order $p = 0, 1, 2$ is then obtained by concatenating all subdictionaries :

$$\boldsymbol{A} = [\boldsymbol{A}_0 | \boldsymbol{A}_1 | \boldsymbol{A}_2] \tag{3.15}$$

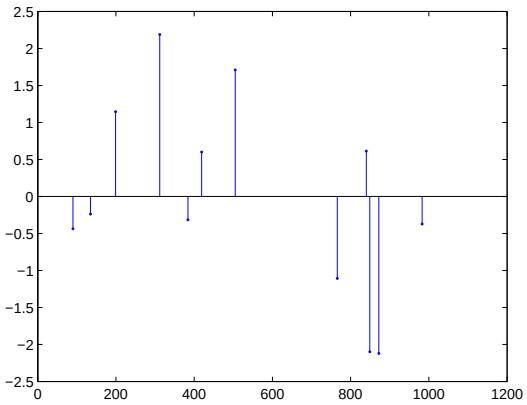
yielding matrix $\boldsymbol{A}$ of size 516×1545.

We run *SolveLasso_Flops* until the maximal sparsity level k reaches 20. Because the outputs are optimizers for ℓ_1-constrained least-square problem, but not optimizer for ℓ_0-constrained least-squares problem, in order to fairly compare the least-square costs of different algorithms, a post-processing is needed to recalculate the least-square error :

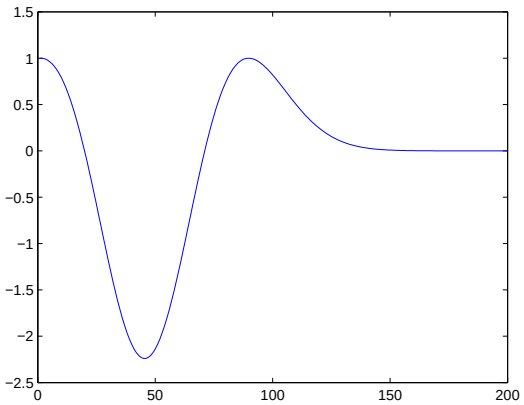
$$\mathcal{E} = \min_{\mathcal{S}(\boldsymbol{x}) \subseteq \mathcal{S}(\tilde{\boldsymbol{x}})} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 \tag{3.16}$$

where $\mathcal{S}(\tilde{\boldsymbol{x}})$ is the support of ℓ_1-regression estimate $\tilde{\boldsymbol{x}}$. Sometimes there exists more than one $\tilde{\boldsymbol{x}}(\lambda)$ with different λ share the same sparsity level, *i.e.,* $\|\tilde{\boldsymbol{x}}(\lambda_1)\|_0 = \|\tilde{\boldsymbol{x}}(\lambda_2)\|_0 = k$, so we choose the one giving the lowest least-square cost as the solution for the given sparsity level k, yielding ℓ_0-constrained solution path $\{\hat{\boldsymbol{x}}_k | k = 1, 2 \ldots, 20\}$. We run the final version of CSBR to estimate a ℓ_0-constrained least-square solution path with sparsity level from 1 to 20. Since *hard_l0_Mterm* works for given sparsity level, we run it 20 times from $k = 1$ to $k = 20$, with $\hat{\boldsymbol{x}}_{k-1}(k > 1)$ and $\hat{\boldsymbol{x}}_0 = \boldsymbol{0}(k = 1)$ as initial vector.

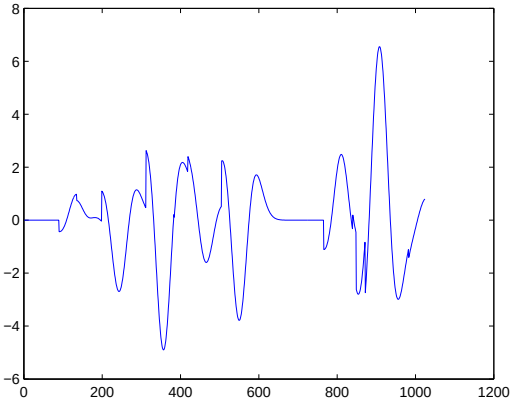
The approximation error (least-square cost $\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}_k\|^2$) *vs* sparsity level k is shown in Fig. 3.2, which can give the user a global idea of the tradeoff profiles. We can see when sparsity level is above $k = 5$, the approximation performance of CSBR outperforms that of the other two algorithms. In Fig. 3.3, we show the reconstructions and detected discontinuities with sparsity level $k = 11$. CSBR provides more meaningful results.



sparse spike train $\boldsymbol{x}_0$

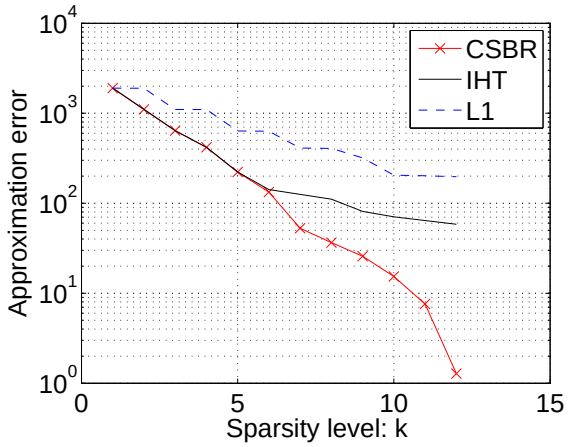


impulse response function $\boldsymbol{h}$



observation $\boldsymbol{y}$

FIGURE 3.4 – Signals in sparse spike train deconvolution problem. $\boldsymbol{x}_0$ is the sparse spike train, $\boldsymbol{h}$ is the impulse response function (only the first 200 samples are shown here, the rest are zero), a truncated second derivative of a Gaussian, $\boldsymbol{y}$ is the observation, *i.e.*, the convolution between $\boldsymbol{x}_0$ and $\boldsymbol{h}$.



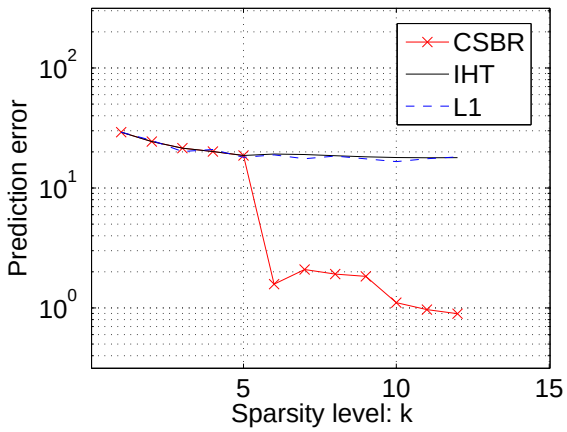


FIGURE 3.5 – Approximation (left) and prediction (right) performance in sparse spike train deconvolution problem.

TABLE 3.3 – Correspondence between mutual coherence $\mu(\boldsymbol{A})$ and radius $\log(r)$ in the mutual coherence sensitivity problem.

$\log(r)$	-1.6	-1.4	-1.2	-1.0	-0.8	-0.6	-0.4	-0.2	0
$\mu(\boldsymbol{A})$	0.9806	0.9543	0.9014	0.8135	0.6979	0.5863	0.4966	0.4427	0.4188

3.4.2 Sparse spike train deconvolution

In Chapter 2, we already tested SBR in a deconvolution problem. In this experiment, we test CSBR in a standard deconvolution problem coming from *Sparco*. *Sparco* [54] is a framework for testing and benchmarking sparse reconstruction. It includes a large collection of sparse reconstruction problems drawn from the imaging, compressed sensing [55], and geophysics literature. In Problem903, a sparse spike train $\boldsymbol{x}_0$ pass a filter whose impulse response function $\boldsymbol{h}$ is a truncated second derivative of a Gaussian, yielding observation $\boldsymbol{y}$ (see Fig. 3.4). Inferring $\boldsymbol{x}_0$ from $\boldsymbol{y}$ and $\boldsymbol{h}$, is known as deconvolution, which is a classical problem in signal processing community.

We formulate this problem as ℓ_0-constrained least-square problem (3.1), the dictionary $\boldsymbol{A}$ is a Toeplitz matrix formed by $\boldsymbol{h}$, and the constraint is $\|\hat{\boldsymbol{x}}\|_0 \leq \|\boldsymbol{x}_0\|_0 = 12$. $\boldsymbol{y}$ is of size 1024×1, $\boldsymbol{H}$ is of size 1024×1024 and $\boldsymbol{x}$ is of size 1024×1. We use CSBR, IHT and homotopy to estimate a solution path ($k = 1, 2, \ldots, 12$). In order to have comparable computation time, we set the maximal iteration *maxIter* to 1000 for each running of *hard_l0_Mterm*. In our experiment, we find that increasing the maximal iteration do not improve the performance. We run the simulation on a Dell Precision T7400 workstation (4 core 3GHz CPU, 3.4 GB memory). CSBR cost 0.39 seconds, IHT cost 1.82 seconds and ℓ_1-regression cost 0.12 seconds.

The approximation error ($\|\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_k\|^2$) *vs* sparsity level and prediction error ($\|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_k\|^2$) *vs* sparsity level are shown in Fig. 3.5. We can see when sparsity level is above $k = 6$, CSBR outperforms ℓ_1-regression and IHT in both approximation and prediction performance. The estimates with sparsity level 12 are shown in Fig. 3.6. CSBR found other 10 spikes correctly except the 2 spikes located at 384 and 419.

3.4.3 Sensitivity to the mutual coherence of the dictionary

In order to compare the performances of algorithms under different difficulty level *i.e.,* mutual coherence and sparsity level, in this simulation, we construct the dictionary $\boldsymbol{A}$ whose mutual coherence $\mu(\boldsymbol{A})$ can be adjusted by the parameter r. The first column of $\boldsymbol{A}$ is fixed as $\boldsymbol{a}_1 = [1, r, 0, \ldots, 0]^T$. The other columns of $\boldsymbol{A}$ are generated as $\boldsymbol{a}_n = \boldsymbol{a}_1 + r\boldsymbol{\nu}(n > 1)$, where $\boldsymbol{\nu}$ is a random vector whose entries have i.i.d. normal distribution. Thus all the columns of $\boldsymbol{A}$ form a ball centered at $\boldsymbol{a}_1$ with radius being controlled by r. By adjusting r, the mutual coherence can be changed. Small r indicates the columns of $\boldsymbol{A}$ are close to each other, yielding large mutual coherence, *vice-versa*. In our experiment, $\log(r)$ is sampled 9 points from -1.6 to 0 with step 0.2. The corresponding mutual coherence is shown in Tab. 3.3, which ranges from 0.98 to 0.42.

The experiments are carried out as : For each matrix $\boldsymbol{A}$ of size 100×200, we test the sparsity level k from 3 to 10. For each given k, we generate 1000 Monte Carlo trials. In each Monte Carlo trial, we generate $\boldsymbol{x}_0$ with k spikes, the locations of the k spikes have uniform distribution on the length of vector $\boldsymbol{x}_0$, and the amplitudes of the k spikes have i.i.d. normal distributions. Each observation is then generated as $\boldsymbol{y}_0 = \boldsymbol{A}\boldsymbol{x}_0$. From each observation $\boldsymbol{y}_0$, we run CSBR and IHT to estimate a solution $\hat{\boldsymbol{x}}_0$ with sparsity level k which is known in advance. If the support of $\hat{\boldsymbol{x}}_0$ is the same as that of $\boldsymbol{x}_0$, we say that this experiment is a **success**. For ℓ_1-regression, we test the support of estimated solution path, *i.e.,* if there exist one solution whose support is the same as that of $\boldsymbol{x}_0$, we say this experiment is a success for ℓ_1-regression. Fig. 3.7 shows the rate of success of the three algorithms.

From Fig. 3.7, in all the regions, CSBR achieves the rate of success close to 1. The performance

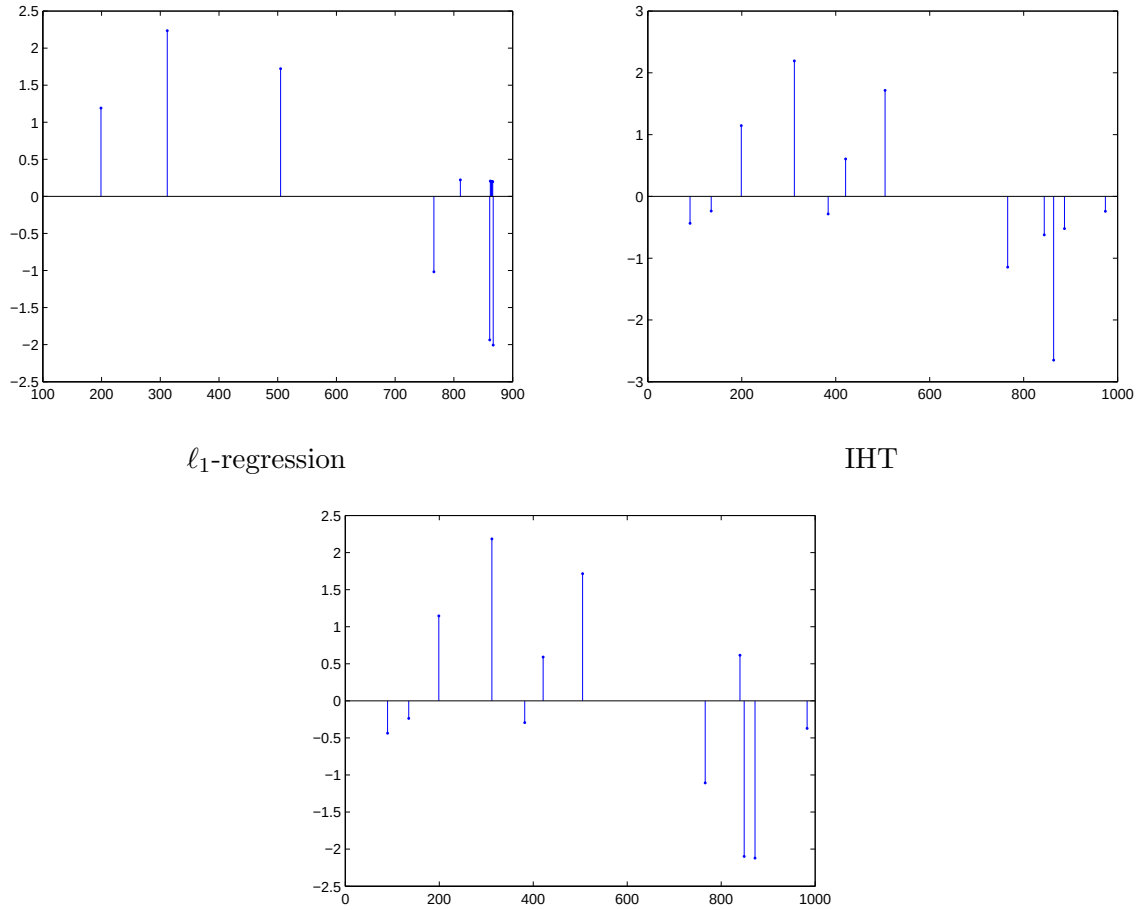


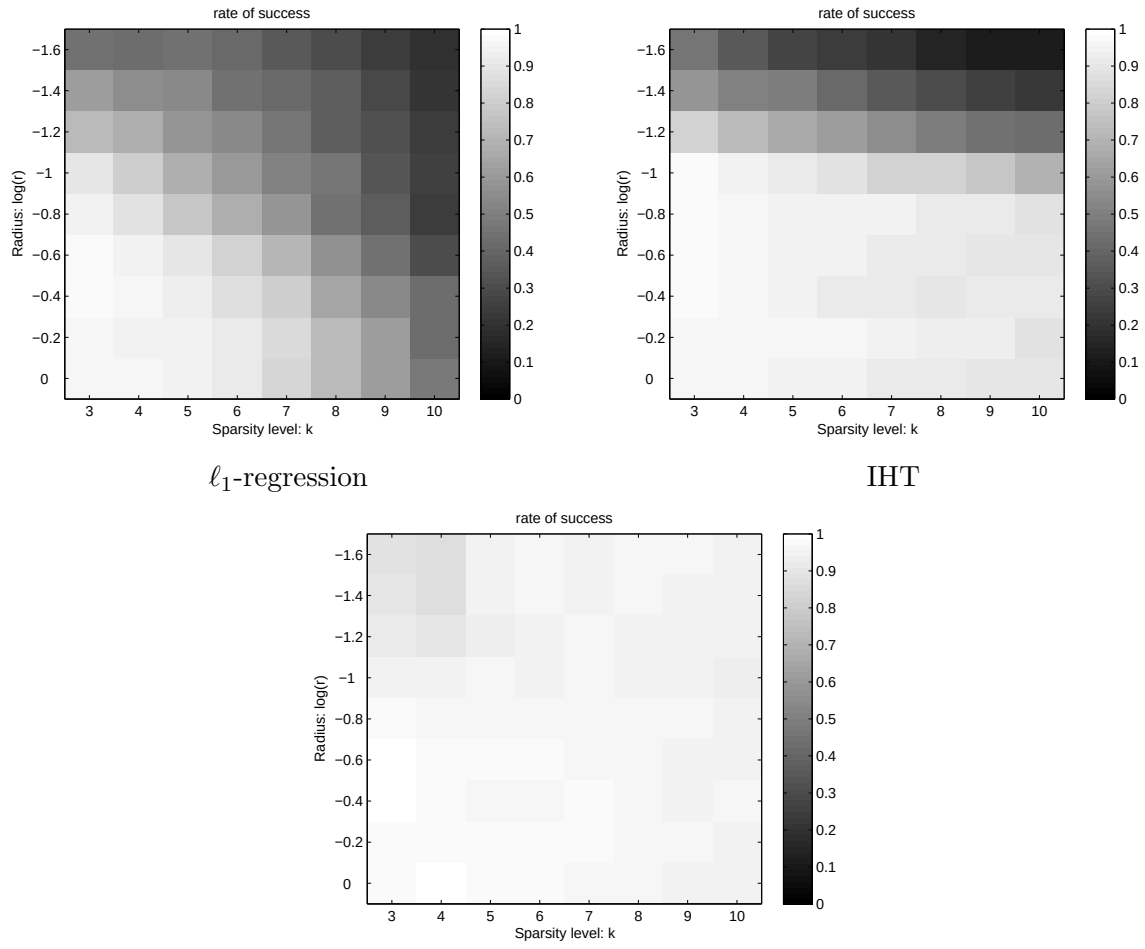
FIGURE 3.6 – Signal estimates $\hat{\boldsymbol{x}}$ in sparse spike train deconvolution problem.

of ℓ_1-regression depends on sparsity level k and radius r. In small k and large radius r region, the performance is quite good, while for large k and small radius r region, ℓ_1 regression does not work at all. For IHT, the performance drastically degenerates when $\log(r)$ cross -1.2, which corresponds mutual coherence $\mu(\boldsymbol{A}) \approx 0.9$. The conclusion is, CSBR outperforms both IHT and ℓ_1-regression in both high k region and high $\mu(\boldsymbol{A})$ region, where the using of CSBR is profitable.

3.5 Conclusion

From the last experiment (see Fig. 3.7), we can see that in the high mutual coherent (low $\log(r)$) and large k region, CSBR outperforms IHT and ℓ_1-regression. In fact, when we calculate the mutual coherence of the first and second experiments, we found that the mutual coherence reach almost 1, so the different performance of CSBR, IHT and ℓ_1-regression is clear. The reason, as already discussed in Chapter 2, is that lots of removals occur. We also compared the three algorithms in a low mutual coherence problem : Problem006 in Sparco, which is a typical compressed sensing problem, using Daubechies basis, Gaussian ensemble measurement matrix to approximate a piecewise cubic polynomial signal. Because the mutual coherence of Gaussian ensemble measurement matrix is rather low (0.2), so CSBR, ℓ_1-regression and IHT yielded the similar performance, the reason is that removals rarely occur in comparison with the insertion.

The choose of hyperparameter λ or k highly influences the detection of discontinuity and the quality of approximation. Instead of trying lots of hyperparameter empirically in order to find a



ℓ_1-regression IHT

CSBR

FIGURE 3.7 – The rate of success of ℓ_1-regression, IHT and CSBR in the mutual coherence sensitivity problem.

reasonable solution, the motivation of developing CSBR is to estimate a solution path, which if of great importance when processing real data, where the real sparsity level and noise variance are unknown. Once the solution path is known, we can display all the tradeoff of approximation error $\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2$ *vs* sparsity level k on a 2D figure (see Fig. 3.2), which can help the user have a global view of the tradeoff profiles. The automatic choice of hyperparameter is also of great interest. Available algorithms are L-curve [56], Akaike Information Criterion (AIC) [57] and Minimum Description Length (MDL).

Chapitre 4

Automated force volume image processing for biological samples

Pavel Polyakov [1#], Charles Soussen[2#], Junbo Duan[2], Jérôme F.L.Duval[3], David Brie[2*] and Grégory Francius[1*]

Abstract : Atomic force microscopy (AFM) has now become a powerful technique for investigating on a molecular level, surface forces, nanomechanical properties of deformable particles, biomolecular interactions, kinetics, and dynamic processes. This paper specifically focuses on the analysis of AFM force curves collected on biological systems, in particular bacteria. The goal is to provide fully automated tools to achieve theoretical interpretation of force curves on the basis of adequate, available physical models. In this respect, we propose two algorithms, one for the processing of approach force curves and another for the quantitative analysis of retraction force curves. In the former, electrostatic interactions prior to contact between AFM probe and bacterium are accounted for and mechanical interactions operating after contact are described in terms of Hertz-Hooke formalism. Retraction force curves are analyzed on the basis of the *Freely Jointed Chain* model. For both algorithms, the quantitative reconstruction of force curves is based on the robust detection of critical points (jumps, changes of slope or changes of curvature) which mark the transitions between the various relevant interactions taking place between the AFM tip and the studied sample during approach and retraction. Once the key regions of separation distance and indentation are detected, the physical parameters describing the relevant interactions operating in these regions are extracted making use of regression procedure for fitting experiments to theory. The flexibility, accuracy and strength of the algorithms are illustrated with the processing of two force-volume images which collect a large set of approach and retraction curves measured on a single biological surface. For each force-volume image, several maps are generated, representing the spatial distribution of the searched physical parameters as estimated for each pixel of the force-volume image.

Keywords : Force spectroscopy analysis, force-volume imaging, elasticity, electrostatics, FJC model, curve fitting, detection of critical points, automated algorithm.

1. Laboratoire de Chimie Physique et Microbiologie pour l'Environnement. LCPME, UMR 7564, Nancy-Université, CNRS. 405 rue de Vandoeuvre, F-54600 Villers-lès-Nancy, France.

2. Centre de Recherche en Automatique de Nancy. CRAN, UMR 7039, Nancy-Université, CNRS. Campus Sciences, B.P. 70239, F-54506 Vandoeuvre-lès-Nancy, France.

3. Laboratoire Environnement et Minéralurgie. LEM, UMR 7569, Nancy-Université, CNRS. B.P. 40, F-54501 Vandoeuvre-lès-Nancy, France.

#. Authors have equally contributed to this work.

∗. Corresponding authors. gregory.francius@lcpme.cnrs-nancy.fr (33) 03 83 68 52 36, david.brie@cran.uhp-nancy.fr (33) 03 83 68 44 74

4.1 Introduction

The physico-chemical characterization of biological materials in general, and bacteria in particular, is an important challenge in domains as diverse as biology, microbiology, pharmaceutic and environmental industry, as well as in the field of clinical medicine. Determination of physico-chemical properties of bacteria in terms of electrostatic charge or elasticity, is of fundamental relevance *e.g.,* for understanding bacterial adhesion and infection processes. In addition, several analyses have now evidenced that external structures at the outer periphery of bacteria, like biopolymers or proteins, characterized by different physico-chemical properties, play distinct roles in numerous physical and biological interfacial processes, *e.g.,* plasmid transfer through conjugation [58], adherence to materials or to host cell surfaces [59], cell-cell interactions [60], biofilm formation [61, 62, 63, 64, 65], mobility [66, 67, 68] and pathogenicity [69, 70, 71, 72].

Atomic force microscopy (AFM) emerged within the last decade into a powerful tool for various physical and biological applications [73, 74]. It can operate under wet or physiological conditions [75, 76] with a spatial resolution as high as ∼0.1 nm. AFM force spectroscopy now allows probing of mechanical properties of soft biological samples [77, 78, 79, 2, 80] and measurement of inter- and intramolecular interactions between biomolecules, thus providing new insights into the molecular bases of macromolecular elasticity [81, 82], protein folding [83], and receptor-ligand interactions [84]. AFM is now regarded as a very suitable technique for investigating processes connected to molecular recognition, and for providing valuable information at a molecular scale on the dynamics of individual ligands and receptors on biosurfaces [85] *via* the analysis of retraction force curves. Mechanical properties of soft samples can be evaluated upon appropriate interpretation of approach force curves, so-called nanoindentation analysis [86, 87, 88] According to the latter, viscoelastic properties of living cells may be recovered [89], *e.g.,* elasticity of human cells [90, 91, 92, 93], bacteria [94, 95, 96], or soft gels [97, 98]. In addition, other physico-chemical parameters like charge density or turgor pressure of bacteria covered by specific proteins, polysaccharides or lipopolysaccharides, are now accessible by AFM [99]. In view of the recent numerous and impressive developments AFM technique has undergone, we may state that it is now a necessary tool for understanding a large spectrum of biochemical and biophysical signals which are of utmost importance in clinical medicine [100, 101, 102], life science [103, 104, 105, 106], environmental science [107], and cosmetic industry [108]. For the sake of further illustration, previous works in nanomedicine have demonstrated that cancer, tumor and stem cell biology are regulated by mechanical properties of cells [109, 110, 111, 112] and some diseases can now be diagnosed with use of AFM [113, 114, 115, 116].

The analysis of complex, heterogeneous biological systems can be performed through force-volume imaging (FVI) in which a set of force curves are being recorded on a spatial grid defined on a given sample surface. Analysis of such FVI aims at providing *in fine* a mapping, *i.e.,* a spatial distribution of the relevant physical parameters pertaining to the biological sample. Because of the large amount of force curves necessary to generate a FVI, there is a critical need to develop robust computational methods to achieve :

- accurate determination of the physical parameters of interest all over the biological sample surface ;
- fast and automated processing of each force curve in the whole FVI.

This is essentially the purpose of this paper. In details, we are interested in determining the electrostatic and mechanical properties of biological particles like bacteria. This is achieved upon quantitative analysis of the approach curve between sample and AFM tip prior to and after contact. For that purpose, the force curves are modeled by a piecewise parametric model (electrostatic interaction, Hertz interaction, Hooke interaction) allowing for the concomitant estimation of the spatial range where electrostatic interactions are operative and the evaluation of the Young modulus and the spring constant of the bacteria. This clearly constitutes a major contribution because the currently available computational methods [117, 118, 119, 120] essentially allow for the estimation of physical parameters pertaining to mechanical interactions only. We also address the automated analysis of

biopolymer uncoiling in the course of retraction of the tip from the sample surface (retraction force curves). The issue is then to perform an accurate interpretation of the non-monotonous retraction force curves by means of *Worm Like Chain* (WLC) or *Freely Jointed Chain* (FJC) models. To the best of our knowledge, analysis of retraction force curves according to these models is commonly done *via* a manual, *ad hoc* and time-consuming specification of the various intervals in spatial separation between AFM tip and sample [120]. We propose here a fully automated method for fitting retraction force curves on the basis of the aforementioned models. Similar to the analysis of the approach curves, the method consists in a piecewise parametric modeling of the force curves recorded upon retraction of the tip from the investigated sample.

When fitting a series of physical models to reconstruct the various interaction regimes pictured by a given force curve, a key issue is the accurate determination of the various zones where one type of interaction is operating, or equivalently the evaluation of the critical points (as also called change points in statistical signal processing) marking the transition between two interactions of different natures, *e.g.,* electrostatic and mechanical interactions upon gradual approach of the AFM tip towards the sample. Another difficulty is that the parametric models necessary to recover the approach force curves do not apply within the same regime of tip-to-sample separation : the electrostatic model indeed essentially depends on the probe-to-sample separation distance while the contact models (Hertz's and Hooke's theories) involve the so-called deformation depth or indentation of the sample. A mandatory prerequisite for appropriate definition of the latter is the accurate localization of the 'contact point'[97, 121]. Many studies have already stressed this important issue, and in particular highlighted that determination of the contact point is a tricky operation for deformable biological samples [117, 118, 122, 123, 124]. This is also true for situations as simple as those requiring the sole application of Hertz model where neither electrostatic interaction nor linear Hooke mechanical deformations are considered [119, 125]. From a numerical point of view, finding critical points requires to solve a *discrete* optimization problem as the indices of the critical data points must be estimated. This is far from being an easy task within a limited computation time. Indeed, a force curve typically includes thousands of data points and exhaustive discrete search algorithms are known to be numerically time-consuming when the search domain is large. Once the critical points are found, fitting of the experimental data to a given physical model within the required spatial range can be carried out by means of standard local continuous optimization algorithms leading to the estimation of the searched key parameters. For a limited number of unknown parameters, running continuous optimization algorithms is a rather fast process, although there can be issues depending on the degree of sophistication of the models (local minimizers, flat cost function) [108].

The paper is organized as follows. First, we outline the physical models used for reconstructing the approach and retraction force curves. The physical models correspond to the different interactions (electrostatic, mechanical and adhesion forces) that take place between biological samples and AFM tip. Then, the computational method employed for extracting the physical parameters of interest is thoroughly described. To illustrate the performance of our analysis and perform a mapping of bacterial cell physical properties (*e.g.,* elasticity, adhesion), we selected two bacterial models. We studied respectively *Escherichia coli* K12 mutant whose mechanical properties are well-known [99], and performed single-molecule force spectroscopy (SMFS) experiments on *Pseudomonas fluorescens* because of its exopolyssacharides production. In the last part of this paper, we thoroughly discuss the results and comment on the benefits of our computational method for the analysis of force-volume images.

4.2 Theory and physical models

The quantitative analysis of a given force curve requires specific models describing the different physical interactions occurring during approach and retraction. In particular, we focus on the elec-

trostatic interaction between *E. coli* and AFM tip during the approach of the tip, and further analyse the subsequent deformation of the sample after contact with the tip. For retraction curves, we use FJC-based models to describe the stretching of *Pseudomonas fluorescens* bacterial polysaccharides [78, 126, 127].

4.2.1 Physical models relevant for the approach curves

In situations where AFM tip and bacterium are gradually brought together, electrostatic interactions between tip and bacterium first take place, followed by mechanical contact and deformation of the cell envelope as a result of the compression exerted by the tip. The underlying force is not measured directly, but evaluated according to Hooke's law :

$$F_{Exp} = k_c d \tag{4.1}$$

where F_{Exp} is the experimental force measured by AFM, d is the deflection of the cantilever and k_c the known spring constant of the cantilever. Three regimes in the measured approach force curve may be considered for analysis and in turn allow estimation of (i) the volume charge density within the soft envelope of the bacterial cell and the Debye layer thickness (κ^{-1}) that pertains to the typical spatial range where electrostatic interactions between sample and tip are operative, (ii) the Young modulus (E) which reflects cell surface elasticity, and (iii) the bacterial spring constant (k_{cell}) that is related to the inner turgor pressure (P_0) of the cell.

First, upon approach of the (silicon nitride) AFM tip toward the bacterium, both being generally negatively charged [99, 128], repulsive electrostatic double layer force F_{Elec} needs to be considered. For sufficiently large separation distances where the electric double layers developed around the charged sample and tip weakly overlap, F_{Elec} may be expressed within the Debye-Hückel approximation as detailed by Ohshima [129]. The expression for F_{Elec} may then always be written in the form [130]

$$F_{Elec} = A exp\left(-\frac{z}{\kappa^{-1}}\right) \tag{4.2}$$

where z is the probe tip to cell surface separation distance, κ^{-1} the Debye layer thickness and the scalar prefactor A is a function of the dielectric constant of the medium ε_r, the surface charge density (or surface potential) of the hard probe tip σ_r, the thickness h and volume charge density of the soft cell envelope. More detailed theory is required to determine the exact mathematical dependence of A on the searched physicochemical properties of the cell envelope, those of the tip being generally known from independent AFM or electrokinetic measurements [131]. It is beyond the scope of the current paper to provide these expressions that are also depending on tip geometry. For the sake of generality, we reason in the following on the basis of the general expression given by (4.2) and focus on the automated determination of the key prefactor A that contains all searched information pertaining to the electrostatic features of the investigated sample. The quantity κ^{-1} is depending on the concentration of electrolyte in solution according to the expression :

$$\kappa^{-1} = \left(\frac{2F^2 c^{\infty}}{\varepsilon_0 \varepsilon_r RT}\right)^{-\frac{1}{2}} \tag{4.3}$$

with F the Faraday constant, R the gas constant, T the temperature, ε_0 the dielectric permittivity of vacuum, ε_r the relative dielectric permittivity of the aqueous medium, c^{∞} is the bulk concentration of the here-considered monovalent electrolyte. The automated analysis described below for obtaining the prefactor A also allows evaluation of κ^{-1}. The consistency of the analysis may then be checked upon comparison of the so-determined κ^{-1} with the theoretical prediction given by (4.3). It is emphasized that for repulsive interactions the prefactor A is positive while for attractive interactions it changes sign. Situations where $A < 0$ may be encountered for interactions of bacteria with

chemically modified AFM tips carrying a positive surface charge [131].

Once the AFM tip and the cell envelope are in contact, mechanical deformation of the cell envelope takes place. In details, one may distinguish non-linear and linear deformation of the cell envelope upon compression by the AFM tip. The bacterial Young modulus E is obtained from quantitative interpretation of the non-linear regime that follows the electrostatic part detailed above. The corresponding interaction force, denoted as F_{Hertz}, is a function of the so-called indentation δ, *i.e.,* the deformation of the bacterial wall. We apply the Hertz model [132] which is relevant for the deformation of a soft planar interface under action of a tip of conical geometry :

$$F_{Hertz} = \frac{2ETan(\alpha)}{\pi(1-\nu^2)}\delta^2 \tag{4.4}$$

where ν is the Poisson coefficient (0.5 in case of incompressible medium) and α is the semi-top angle of the tip (35^o-value given by manufacturer). The bacterial spring constant value k_{cell} is related to the slope of the linear part of the force *versus* indentation curve that follows the aforementioned non-linear regime [133] :

$$F_{Hooke} = k_{cell}\delta \tag{4.5}$$

The obtained value of k_{cell} is related to the value of the turgor pressure of the cell as detailed in Supplementary Information.

4.2.2 Physical models relevant for the retraction curves

In retraction experiments, biomacromolecules located on the surface of the biological sample are stretched upon removal of the chemically modified AFM tip away from the surface. The obtained force *versus* distance curves are then analyzed according to FJC models [134, 135, 136]. This choice is justified because the studied bacterial model (*P. fluorescens*) produces exopolysaccharides, as suggested by recent FTIR measurements (data not shown, paper in preparation). In addition, FJC models are the most frequently used in the literature [78, 127, 136, 137, 138] to describe the polysaccharide extension behavior because polysaccharide monomers can be regarded as independent rotating segments unlike protein constituents for which WLC models are more suitable [139]. More details are given in the 'Results and Discussion' section for the preparation of the bacterial strains investigated here.

Within the framework of the FJC model, it is assumed that the macromolecule consists of rigid segments connected through flexible joints. The extension z of the macromolecule may then be expressed as a function of the pulling force F *via* the expression [134, 135] :

$$z_{FJC} = -L_c\left[coth\left(\frac{Fl_k}{k_bT}\right) - \frac{k_bT}{Fl_k}\right] \tag{4.6}$$

where the Kuhn length l_k is a direct measure of the chain stiffness, L_c is the total contour length of the macromolecule and k_b is the Boltzmann constant. In the extended FJC+ model, the segments are being assimilated to elastic springs characterized by a segment elasticity k_s. The expression for the macromolecule extension then becomes : [134, 135]

$$z_{FJC+} = -L_c\left[coth\left(\frac{Fl_k}{k_bT}\right) - \frac{k_bT}{Fl_k}\right]\left(1 + \frac{F}{k_sL_c}\right) \tag{4.7}$$

Notice that the number of monomers N in the biomacromolecule is simply related to L_c and l_k according to $N = L_c/l_k$.

For the sake of simplicity, we choose to mainly consider the FJC model within the algorithm described below. The flexibility of the algorithm to generate data fitting according to the FJC+

model will be discussed in the 'Results and Discussion' section.

4.3 Algorithms

For both approach and retraction cases, the models introduced above are *piecewise models* for force curves, *i.e.,* they are strictly applicable within well-defined (but unknown *a priori*) regions of separation distance z, indentation δ or macromolecular extension. In the approach case, the electrostatic regime applies until the contact point is reached. Then, after contact, the Hertz and Hooke regimes successively take place upon gradual compression of the bacterial cell by the tip. In the retraction case, expressions derived from the FJC formalism can be viewed as a piecewise equation : the unfolding of molecules leads indeed to a succession of z-intervals where the FJC model holds for quantitative interpretation.

A key prerequisite for the quantitative analysis of a force curve (*i.e.,* for the estimation of the relevant physical parameters) is the accurate identification of the various intervals where each model may be applied, and corresponding regression procedures performed. These intervals, or regions, are illustrated on Fig. 4.1b, 4.1c and 4.1d in the case of approach and retraction curves, respectively. In the following, they will be referred to as *regions of interest*. Their identification relies on a *force curve segmentation* procedure allowing for the detection of a set of *critical points* (or discontinuity points : jumps, change of slope or change of curvature). The proposed algorithm decomposes the problem into three steps :

1. Force curve segmentation ;
2. Detection of the regions of interest where modeling may be carried out ;
3. Fitting of the data to the physical models in each region.

The first step is common to the processing of approach and retraction curves. It will be described first. On the contrary, the two other steps differ because they rely on distinct physical models for the approach and retraction cases. For readability reasons, we will describe the processing steps 2 and 3 for the approach force curves first, and then for the retraction case.

4.3.1 Force curve segmentation

The force curve segmentation procedure consists in the following two tasks :
– Detection of the discontinuity points in the given measured signal ;
– Piecewise-smooth approximation of the signal.
The notations are defined in Fig. 4.1a. z_i and F_i stand for the i-th experimental measurement (the i-th data point in the force curve). The z-axis is defined in such a way that the z_i values are increasing with increasing i : the probe-to-sample contact corresponds to the largest z-value in the approach curve and to the lowest z-value for the retraction curve. Note that in the approach case, the electrostatic force F_{Elec} is now increasing with z, thus (4.2) rereads $F_{Elec} = Aexp\left(\frac{z}{\kappa^{-1}}\right)$.

Let $D_1, \ldots, D_k$ refer to the (unknown) discontinuity points, leading to a series of contiguous intervals $[D_0, D_1)$, $[D_1, D_2)$, …, $[D_{k-1}, D_k)$, $[D_k, z_n)$ with $D_0 \triangleq z_1$ and z_n the minimal and maximal z_i values. Each interval is left-closed and right-opened (except for the last one) so that their union provides the whole interval $[z_1, z_n]$ (see Fig. 4.1a). On each interval, the experimental data are smoothed in the following way. The force measurements F_i corresponding to $z_i \in [D_j, D_{j+1})$, are approximated by a polynomial $F(z; \boldsymbol{a}_j) = \sum_{l=1}^{r} a_j^l z^l$. The approximation is done in the least-squares sense by minimizing the squared error $\sum_{z_i \in [D_j, D_{j+1})} (F_i - F(z_i; \boldsymbol{a}_j))^2$.

The segmentation algorithm is inspired by the Orthogonal Least Squares algorithm [23]. This procedure gradually approximates a given signal upon selection of an increasing number of points from a predefined set of elementary signals. Basically, the segmentation algorithm searches for k

discontinuity points for which the sum of the squared errors $\sum_i \sum_{z_i \in [D_j, D_{j+1})} (F_i - F(z_i; \boldsymbol{a}_j))^2$ is minimal. It is based on the iterative addition of one element into the list of discontinuity positions. In the first iteration, this list is empty. The algorithm then sequentially includes a new discontinuity point and updates the piecewise-smooth approximation until k discontinuities are found. An important issue is the setting of the number of discontinuity points. A first possibility is to define a maximal number of points upon visual inspection of the experimental signal and simple count of the number of discontinuity points. An alternative and more automated procedure consists in setting a threshold value E_{min} for the mean approximation error, *i.e.,* the average of the squared error $(F_i - F(z_i; \boldsymbol{a}_j))^2$ for all i and intervals j. If the mean approximation error is lower than E_{min}, then the algorithm is stopped. The reader is referred to a more formal and detailed description of the segmentation algorithm in a technical report reported in Ref. [53]. In addition, more details are given in the 'Results and discussion' section on the setting of the number of discontinuity points. The segmentation algorithm is illustrated in Fig. 4.2 for the approach and retraction cases.

4.3.2 Fitting of the approach curves

In this section, we describe in details the two remaining steps for quantitative analysis of an approach curve. Let us first introduce formal notations for the critical points marking the onset and the end of the electrostatic interaction in the z-domain. To distinguish them with the data measurements z_i, we use capital letters Z_0 and Z_1 (see Fig. 4.1b). We define the point Z_0 as the z-value below which instrumental limitation renders impossible any accurate measurement of interaction force. Z_0 can be seen as a 'virtual pre-contact point'. It does not correspond to a physical change point nor to a real contact between the AFM tip and the sample. However, it is necessary to consider such arbitrary position to define the indentation δ. The δ-values are defined according to $\delta = z - Z_0 - (F_{Exp} - F_0)/k_c$ where F_0 is the force value at $z = Z_0$ ($\delta = 0$ for $z = Z_0$). The next critical point $z = Z_1$ corresponds to the transition from the electrostatic to the Hertz regimes (Fig. 4.1b). Because the non-linear and linear mechanical deformations of the cell wall are defined in the indentation domain, the transition from the Hertz to the Hooke regime is defined accordingly using the quantity δ. This position, denoted as Δ_2, will be estimated in the regression procedure described below. In brief,

- for $z \leq Z_1$, the electrostatic regime holds ; the contact point (Z_1) has not been reached ;
- for $z \geq Z_1$ and $\delta \leq \Delta_2$, the Hertz regime holds ;
- for $z \geq Z_1$ and $\delta \geq \Delta_2$, the Hooke regime holds.

In the second step of the algorithm, the regions of interest $Z_0 < z_i \leq Z_1$ (electrostatic regime) and $z_i > Z_1$ (Hertz and Hooke regimes) are detected based on the segmentation results. The discontinuity points provided by the segmentation algorithm are used in the following manner. Let Z_{0+} be the first data point satisfying the condition $F_i \geq 0$ for all $z_i \geq Z_{0+}$. Z_0 and Z_1 are then defined as the two *consecutive* discontinuity points $Z_0 = D_j$ and $Z_1 = D_{j+1}$ surrounding Z_{0+} ($D_j \leq Z_{0+} \leq D_{j+1}$). The detection of the electrostatic region is illustrated on Fig. 4.2b where the dashed arrows represent the critical points Z_0 and Z_1. Also, the following discontinuity point D_{j+2} is used as an initial coarse estimation of the edge between the Hertz and Hooke regions in the z-domain : $Z_2^{init} = D_{j+2}$. On Fig. 4.2b, this position is represented with a plain arrow. Finally, the accurate (final) estimation of this edge position (Δ_2) is done in the δ-domain.

Once both electrostatic and contact regions are clearly identified, the remaining task (third step) consists in a fit of the data to the appropriate physical models. The unknown parameters are estimated in the least-squares sense, *i.e.,* we minimize the cost function defined as the sum of the squared errors between experimental data and their approximation from the parametric model. We now define two cost functions corresponding to the electrostatic and mechanical interactions. In each case, we detail the imposed constraints on the parameter values. Because we use a local gradient-based algorithm whose behavior can depend on the setting of the initial parameter values, we also detail the initialization step and propose heuristic rules for each model.

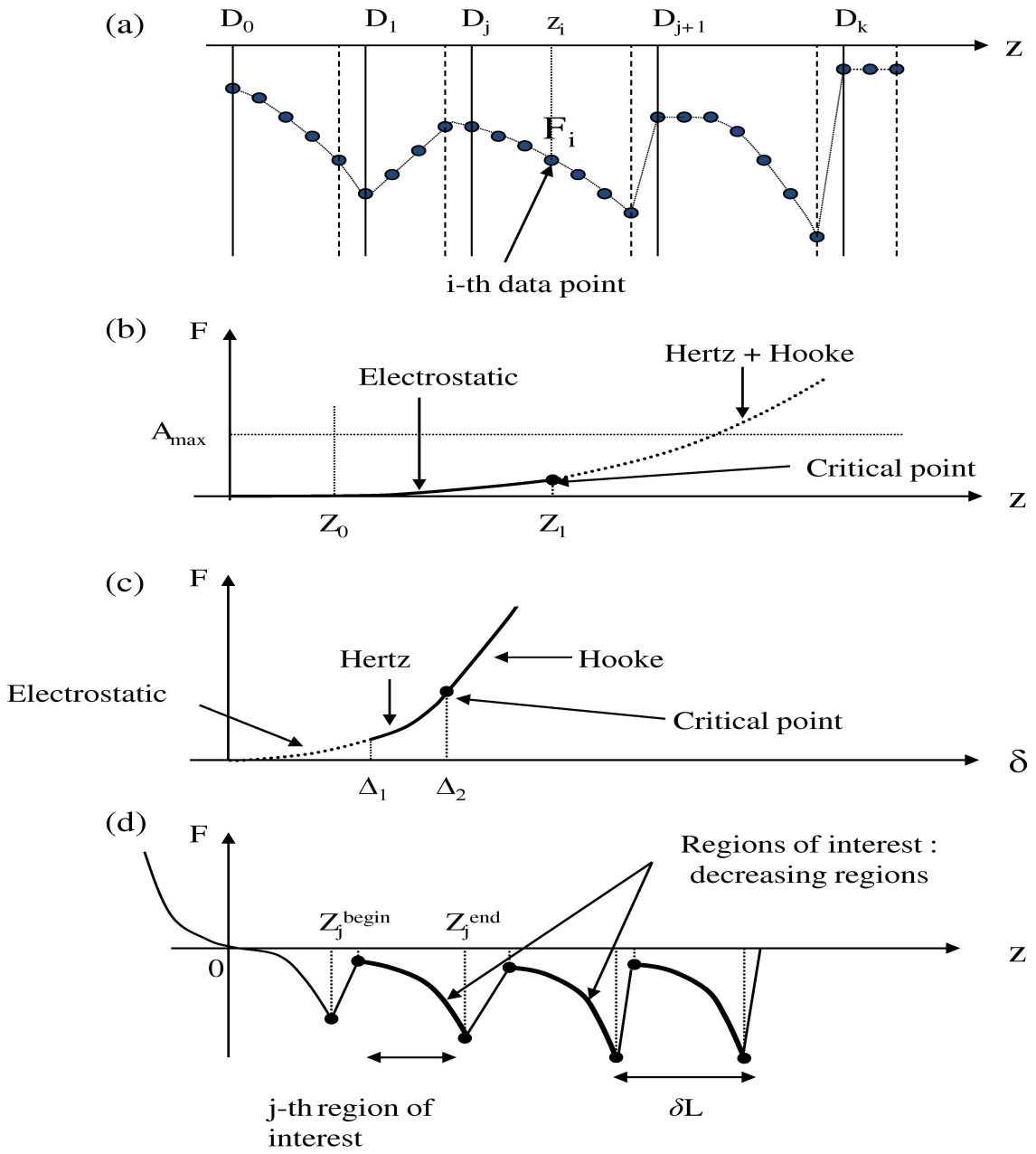


FIGURE 4.1 – Main notations and schematic illustration for the three steps of the proposed algorithm. (a) Segmentation of a force curve (first step). The data points (z_i, F_i), $i = 1, \ldots, n$ are partitioned into $k+1$ contiguous intervals $[D_0, D_1), [D_1, D_2), \ldots, [D_{k-1}, D_k), [D_k, z_n]$ with $D_0 \triangleq z_1$ and z_n the minimal and maximal z_i values. The plain and dashed vertical lines represent the left and right bounds of the segmented intervals. The experimental data F_i are smoothed on each interval (piecewise smoothing). (b) Detection of the electrostatic region in the z -domain (second step) and fitting to the electrostatic model (third step). The critical points $z = Z_0$ and $z = Z_1$ are the edges of electrostatic region in the z-domain. A_{max} is an upper bound for the exponential prefactor A. It is optional but useful for improving the detection of the edge $z = Z_1$ by forbidding z-values that lead to unrealistic values of A. (c) Segmentation and fitting in the δ-domain (third step). The critical points $\delta = \Delta_1$ and $\delta = \Delta_2$ correspond to the beginning of the Hertzian and Hooke regimes, respectively. (d) Retraction curve : the j-th region of interest is the j-th decreasing interval $z \in [Z_j^{begin}, Z_j^{end}]$. The search for these regions is done from the outputs of the segmentation algorithm (second step), then the data are fitted to the FJC model within each region of interest (third step).

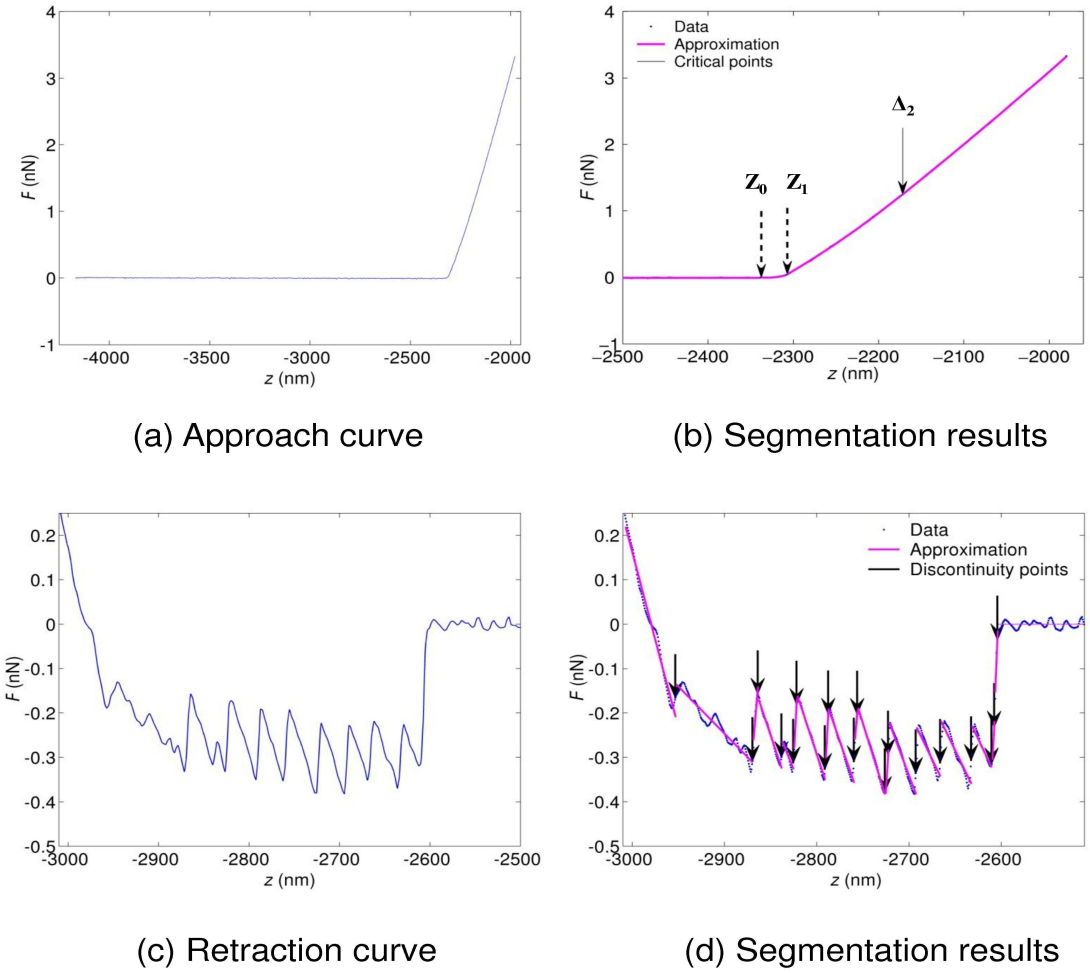


FIGURE 4.2 – Force curve segmentation. The approach and retraction curves correspond to the marked pixel in the *E. coli* cell image shown in Fig. 4.4c. (a) Approach curve : experimental data. The curve is preprocessed by subtraction of an affine baseline. (b) Segmentation results with 6 discontinuity points. In the interval $z \in [-2500, -2000]$ nm, only three discontinuity points are present. The two points represented with dashed arrows are the edges Z_0 and Z_1 of the electrostatic region. The next discontinuity point ($z \approx$-2170 nm) marked by a plain arrow is used as initial estimation of the transition between Hertz and Hooke regimes. (c-d) Retraction curve. The polynomial degree is set to $r = 1$, leading to a piecewise affine approximation of the original signal. The threshold value E_{min} for the mean approximation error is set to the empirical noise variance, and 20 discontinuities at most are detected. For the displayed retraction curve, 20 iterations were performed leading to 20 discontinuities.

In the z-domain, the electrostatic model $F_{Elec}(z;\kappa^{-1},A)$ holds for $z \leq Z_1$ (see (2)). The fitting of the experimental data (z_i, F_i) for $Z_0 \leq z_i \leq Z_1$ is formulated as the minimization of the cost function :

$$K_{Elec}(\kappa^{-1},A) = \sum_{Z_0 \leq z_i \leq Z_1} \big(F_i - F_{Elec}(z_i;\kappa^{-1},A,Z_1)\big)^2 \tag{7}$$

under the constraints $A \geq 0$ (repulsive interactions) and $0 < \kappa^{-1} \leq \kappa_{max}^{-1}$ (see 'Results and Discussion' section for practical setting of κ_{max}^{-1}). The dependence of $K_{Elec}(\kappa^{-1},A)$ on A is quadratic. Therefore, a closed-form expression for A can be derived from κ^{-1}, and the minimization of (9) then reduces to a 1D minimization problem $K_{Elec}(\kappa^{-1},A(\kappa^{-1}))$.

In the δ-domain, the contact (Hertz and Hooke interactions) model reads :

$$F_{Contact}(\delta;\Delta_0,\Delta_2,B,k_{cell}) = \begin{cases} B(\delta-\Delta_0)^2, & \text{if } \Delta_1 \leq \delta \leq \Delta_2\,; \\ B(\Delta_2-\Delta_0)^2 + k_{cell}(\delta-\Delta_2), & \text{if } \delta \geq \Delta_2. \end{cases} \tag{8}$$

where the centering around Δ_0 is introduced so as to take into account an unknown 'reference/virtual' pre-contact point. Δ_1 is set to $\delta(Z_1) = Z_1 - F_{Exp}(Z_1)/k_c$ and Δ_2 is unknown. The estimation of the four parameters Δ_0, Δ_2, B and k_{cell} is done by computing the indentation $\delta_i = z_i - F_i/k_c$ for all $z_i > Z_1$ and by minimizing

$$K_{Contact}(\Delta_0,\Delta_2,B,k_{cell}) = \sum_{z_i \geq Z_1} \left(F_i - F_{Contact}(\delta_i;\Delta_0,\Delta_2,B,k_{cell})\right)^2 \tag{9}$$

under the constraints $0 \leq \Delta_0 \leq \Delta_1$ and $\Delta_1 \leq \Delta_2 \leq \delta_{max}$ where δ_{max} stands for the maximum of all δ_i-values. Since the dependence of $K_{Contact}(\delta_0,\delta_2,B,k_{cell})$ on (B,k_{cell}) is quadratic, the minimization of (9) simplifies into a 2D problem, as B and k_{cell} may be expressed as a function of Δ_0 and Δ_2 according to a closed-form relationship.

The cost functions K_{Elec} and $K_{Contact}$ may contain 'flat valleys'. Also, there may be several local minimizers. This makes the optimization algorithm sensitive to the choice of the initial parameter values (as already mentioned for the electrostatic fitting problem [118]). Thus, it is critical to initialize the algorithm with physically realistic values. Here, we detail the initialization of each parameter.

- For the electrostatic model, the initial value of κ^{-1} is set having in mind the relationship $\kappa^{-1} = F_{Elec}(Z_1;\kappa^{-1},A,Z_1)/F'_{Elec}(Z_1;\kappa^{-1},A,Z_1)$, where F' stands for the derivative of F with respect to z. In practice, we use the smoothed polynomial signal obtained as output of the segmentation algorithm and we compute the ratio between the polynomial value and its derivative at $z = Z_1$. As previously mentioned, A directly derives from κ^{-1} using the closed form expression that relates the two parameters.
- During the segmentation procedure, a rough estimation for the edge between the Hertzian and Hooke regions, Z_2^{init}, is computed. This estimation leads to an initial evaluation of Δ_2 : $\Delta_2^{init} = Z_2^{init} - Z_0 - (F_2^{init} - F_0)/k_c$ with $F_2^{init} = F_{Exp}(Z_2^{init})$. We set the initial value for Δ_0 to 0.

An illustration of the electrostatic fitting is displayed in Fig. 4.3a while the fitting results pertaining to the contact part of the force curve, including the estimation of the Δ_0 and Δ_2 positions, are shown in Fig. 4.3b.

4.3.3 Fitting of the retraction curves

Similar to the approach case, we now detail for the retraction curve the two remaining steps of the fitting algorithm given the outputs D_j of the segmentation procedure.

There are as many regions of interest as the number of segments in the freely jointed chain.

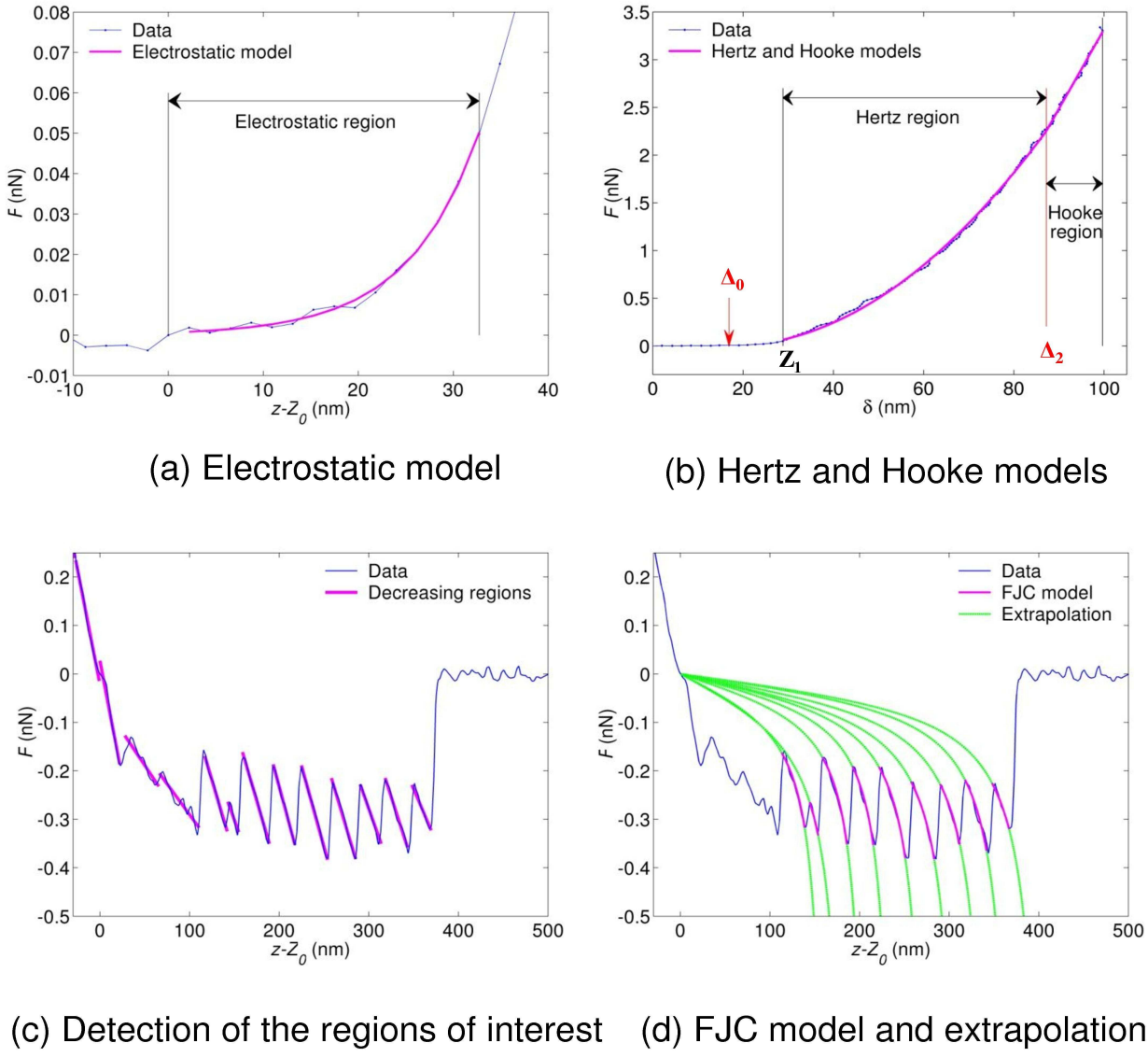


FIGURE 4.3 – Detection of the regions of interest from the segmentation results and fitting to the physical models.(a) Approach curve : both vertical lines correspond to the beginning ($z = Z_0$) and end ($z = Z_1$) of the electrostatic regime. (b) Approach curve, Hertz and Hooke regimes. The red vertical line and red arrow correspond to the positions Δ_0 and Δ_2 which are found by the optimization algorithm. The black line in between marks the δ-value Δ_1 for $z = Z_1$ (beginning of Hertzian regime). (c) Retraction curve : detection of the regions of interest (decreasing regions of the piecewise affine signal). (d) Fitting of the data to the FJC model in each region of interest. No fitting is performed in the first 100 nm after the contact point $z = Z_0$. For each region, the extrapolation of the FJC model outside the current region of interest is shown in green.

Contrary to the approach model, the regions of interest $z \in [Z^{begin}, Z^{end}]$ are not contiguous : two consecutive regions may be separated as indicated in Fig. 4.1d. These regions are denoted as $\mathcal{R}_0, \mathcal{R}_1, \ldots$ and generally differ from the intervals $[D_j, D_{j+1})$ provided by the segmentation algorithm. We now detail their detection according to heuristic rules. Because the FJC curve (seen as a function of $z : z \mapsto F(z)$) decreases in all intervals of interest, we basically consider the intervals $[D_j, D_{j+1})$ where the polynomial approximation $F(z; \boldsymbol{a}_j)$ is a decreasing function. The FJC model (4.6) then applies between $Z^{begin} = D_j$ and $Z^{end} = D_{j+1}$. The detection of the regions of interest is illustrated in Fig. 4.3c together with the piecewise smoothing (piecewise affine smoothing, obtained with polynomial degree $r = 1$). The contact point Z_0 is defined from the corresponding approach curve as detailed above. When the polynomial degree is set to $r = 2$, $F(z; \boldsymbol{a}_j)$ may not be monotonous within the interval $[D_j, D_{j+1})$. For such situations, we split the interval into two sub-intervals where $F(z; \boldsymbol{a}_j)$ is monotonous, and solely consider the sub-interval $[Z^{begin}, Z^{end}]$ where the quadratic polynomial is a decreasing function. Finally, intervals $[Z^{begin}, Z^{end}]$ with $Z^{begin} \leq Z_0$ and intervals containing very few (typically, less than 5) data points are not considered.

For each region of interest $\mathcal{R}_j$, the FJC parameters are the contour length L_c^j and the Kuhn length l_k^j. Their recovery is carried out by fitting the data $F_i(z_i \in \mathcal{R}_j)$ to the FJC model. The least-squares formulation leads to the minimization of

$$K_{FJC}^j(L_c, l_k) = \sum_{z_i \in \mathcal{R}_j} (z_i - z_{FJC}(F_i; L_c, l_k))^2 \tag{10}$$

under the constraints $L_c \geq L_c^{min} \triangleq max(Z_{end}^j, l_k)$ and $l_k \geq 0$ where Z_{end}^j denotes the upper bound of the j-th region of interest. The minimization of (10) is a 2D optimization problem quadratic in L_c. It thus simplifies into the 1D problem $\min K_{FJC}^j(L_c(l_k), l_k)$ subject to $l_k \geq 0$, where for a given l_k, $L_c(l_k)$ stands for the minimizer of $L_c \mapsto K_{FJC}^j(L_c, l_k)$ subject to the $L_c \geq L_c^{min}$ which has a closed-form expression.

We noted that the 1D cost function $l_k \mapsto K_{FJC}^j(L_c(l_k), l_k)$ exhibits large variations for small values of l_k (convex valleys) and becomes very flat for larger l_k. When setting an initial solution of the same order of magnitude as the z_i-values (*e.g.,* $l_k^{init} = Z_{end}^j$), l_k^{init} is found to lay within the flat valley of the cost function, and the local optimization algorithm then often fails to find the global minimizer of the cost function. In practice, we rather set the initial condition to $l_k^{init} = Z_{end}^j/1000$ to ensure that l_k^{init} lays in the non flat convex valley, and close enough to the global minimizer of K_{FJC}^j. In the 'Results and Discussion' section, we discuss the issues relative to fitting to the extended FJC+ model.

4.3.4 Processing of a force-volume image

In this section, we apply the fitting procedure to the recovery of a force-volume image. Each force curve corresponds to a single pixel in the FVI. All force curves are processed sequentially as detailed above, thus providing as many outputs (physical parameters) as there are pixels. Each physical parameter can then be displayed on a 2D map. We emphasize that each force curve is processed using the same design parameters, which results in a highly automated FVI fitting algorithm.

4.4 Results and Discussion

The experimental part of this work has been carried out to illustrate the different steps of the data processing and to verify the coherence of the so-estimated physical parameters. This section is organized as follows. First, we introduce the studied bacteria and describe the sample preparation

procedure. Next, we detail the conditions of data acquisition with the AFM instrument.For both approach and retraction cases, we first illustrate the behavior of the proposed algorithm on a single force curve, and then, we provide results (2D maps representing the physical parameters of interest) obtained by processing a FVI. In particular, westrongly underline that the proposed algorithm is run on the whole FVI using a common parameter setting. We analyze the results and in particular discuss the accuracy of the obtained physical parameters, and compare their values with data reported in literature. Finally, we comment on the behavior and performance of the proposed algorithm and we qualitatively elaborate potential adaptations to other physical models.

4.4.1 Bacterial culture and sample preparation

The first bacterial model used in this study is a gram negative *Escherichia coli* K-12 mutant called (E2152) kindly provided by the Institut Pasteur of Paris. This bacterium was constructed from *E. coli* MG1655 (*E. coli* genetic stock center CGSC#6300) by transformation and λ red linear DNA gene inactivation method followed by P1*vir* transduction into a fresh *E. coli* background when possible. E2152 is characterized by a simple gram negative bacterial cell wall that does not exhibit any biopolymers or external structures at its outer periphery [124]. To extract relevant electrostatic and mechanical properties, we analyze approach force curves collected between the bacterium and standard silicon nitride AFM tip.

Secondly, we analyze retraction force curves pertaining to the uncoiling of exobiopolymers (probably glycogene) from *Pseudomonas fluorescens* using single-molecule force spectroscopy (SMFS). *P. fluorescens* is a common bacterium present in drinking water distribution network [140] and can form biofilms [141] and produce exopolymers [142]. In this analysis, AFM tips are functionalized with *Concanavalin A* lectin in order to detect mannosyl and glucosyl residues present in the bacterial exopolymers (EPS) [138].

4.4.2 AFM measurements and preparation of experiments

AFM images and force-distance curves were recorded using an MFP3D-BIO instrument (Asylum Research Technology, Atomic Force F&E GmbH, Mannheim, Germany). Silicon nitride cantilevers of conical shape were purchased from Veeco (MLCT-AUNM, Veeco Instruments SAS, Dourdan, France), and their spring constants were determined using the thermal calibration method [143], providing values of $\sim$10.4 $\pm$ 1.7 pN/nm. Prior to each experiment, the geometry of the tip was systematically controlled using a commercial grid for 3-D visualization (TGT1, NT-MTD Compagny, Moscow, Russia) and curvature of the tip in its extremity was found to lie in the range $\sim$ 20 to 50 nm. Experiments were performed in 1 mM potassium nitrate solution at pH $\sim$ 6.6 and room temperature. Because previous studies pointed out the possible removal and/or shortening of bacterial appendages upon sample centrifugation [144], bacterial cultures were used without any particular conditioning regarding AFM experiments. Cells were electrostatically immobilized onto polyethyleneimine (PEI)-coated glass slides according to a procedure detailed elsewhere [145]. Such method avoids the necessity to resort to chemical binders between substrate and bacterial sample, thus minimizing any chemical modification of bacterial cell wall/surface organization.

Glass slides were freshly prepared upon immersion in 0.2% PEI solution for 30 minutes, extensively rinsed with Milli-Q water, dried with nitrogen and stored in a sterile Petri dish. One mL of bacterial culture ($OD_{600nm} \sim$ 0.5-0.6) was directly deposited onto the PEI-coated glass slide for 20 minutes and then the bacteria-coated surface was extensively rinsed 3 times with Milli-Q water. Following this step, the sample was immediately transferred into the AFM liquid cell with addition of 2 ml of KNO_3 solution of adjusted concentration and pH 6.6 for imaging and nanomechanical analysis. Single Molecule Force experiments were performed with 2 mL of solution of special buffer (40 mM maleic acid, 60 mM TRIS, 2 mM $CaCl_2$ and 2 mM $MnCl_2$ at pH 5) and AFM tips were

functionalized with *Concanavalin A* [138].

4.4.3 Data processing for the approach force curve

The 512-by-512 pixels AFM image reported in Fig. 4.4a reveals that *E. coli* E 2152 can be assimilated to 5μm long rod-shaped cell, being in the process of division. The horizontal line corresponds to the lateral cross section depicted in Fig. 4.4b. The width ($\sim$ 1.6 μm) of the cell was found to be larger than the height ($\sim$ 0.7 μm), which is essentially caused by artifactual features connected to probe geometry. In order to perform the force volume measurements, a 5μm$\times$5μm scan region was divided into a 32-by-32 grid. For each pixel on this grid, force curves were recorded upon approach of the probe toward the sample surface, using an applied force of 4 nN.

Prior to processing a force-volume image, it is necessary to set the values of the algorithm parameters. An important point is that these parameters have common values for all the force curves. First, a choice must be done regarding the polynomial degree r (for piecewise polynomial smoothing). We set $r = 2$ since approach force curves involve linear and convex regions which can be locally approximated by affine and quadratic polynomials. The choice for the number of iterations of the segmentation step, *i.e.,* the number of discontinuity points, is critical. In the 'Algorithm' section, we introduced two alternative strategies where the number of iterations is maintained constant, or varies according to the data. When processing the approach curves from a whole force-volume image, we use the first strategy since the number of discontinuity points is always limited and does not strongly vary from one curve to another. We observed that the approach curves are very well approximated with 6 discontinuity points. Fig. 4.2a and 4.2b illustrate the segmentation results obtained for an approach curve measured on E2152 bacterial strain (marked pixel of Fig. 4.4c). In details, the approach curve is first pre-processed by removing an affine baseline (typically, the first 500 data points upon approach of the tip). The preprocessed curve is displayed in Fig. 4.2a and the force curve segmentation result (zoom in) is shown in Fig. 4.2b. Fig. 4.3a shows the exponential part of the same force curve while Fig. 4.3b displays this force curve as a function of the indentation δ together with the theoretical fittings in the Hertzian and Hooke regions. Both red line and arrow correspond to the positions Δ_0 and Δ_2 which are found by the optimization algorithm, the latter marking the transition between the Hertzian and Hooke regimes. The black line in between is the δ-value Δ_1 at $z = Z_1$, marking the transition from the electrostatic to the Hertzian regime.

Let us now comment on the results obtained when processing all force curves, each measured for a given pixel of the FVI. The first way to address the quality of the data processing consists in comparing the topographical reconstruction achieved via estimation of the Z_1 values for all pixels with the topography obtained via contact mode measurements. Fig. 4.4b shows the cross section of the bacterial height in contact mode. In Fig. 4.4c, the topographical reconstruction obtained from the processing of the FVI is given, and Fig. 4.4d shows a cross section of this reconstruction. For each pixel, the height corresponds to the Z_1 value, the 0-height being defined as the maximum value of Z_1 over the whole image. Despite the low resolution of the FVI, the topographic reconstruction of the bacterium is in very good agreement with that of the high resolved AFM image of Fig. 4.4a. This is mainly due to the strong effect of particle curvature on AFM reconstruction via contact mode measurements while FVI data processing does not suffer from such limitation connected to convolution issue.

Fig. 4.5a, 4.5c, 4.5e and 4.5g display the 2D maps of the obtained physico-chemical properties : pre-exponential factor A (panel a), Debye length κ^{-1} (panel c), Young modulus E (panel e) and bacterial spring constant k_{cell} (panel g) extracted from the exponential and Hertz-Hooke regions of the measured force curves. The spatial distribution of the bacterial physical properties highlight that whatever the considered physical parameter, maximum values are located on the central part of the bacterium whereas the lower values are located on the edges. This spatial heterogeneity is

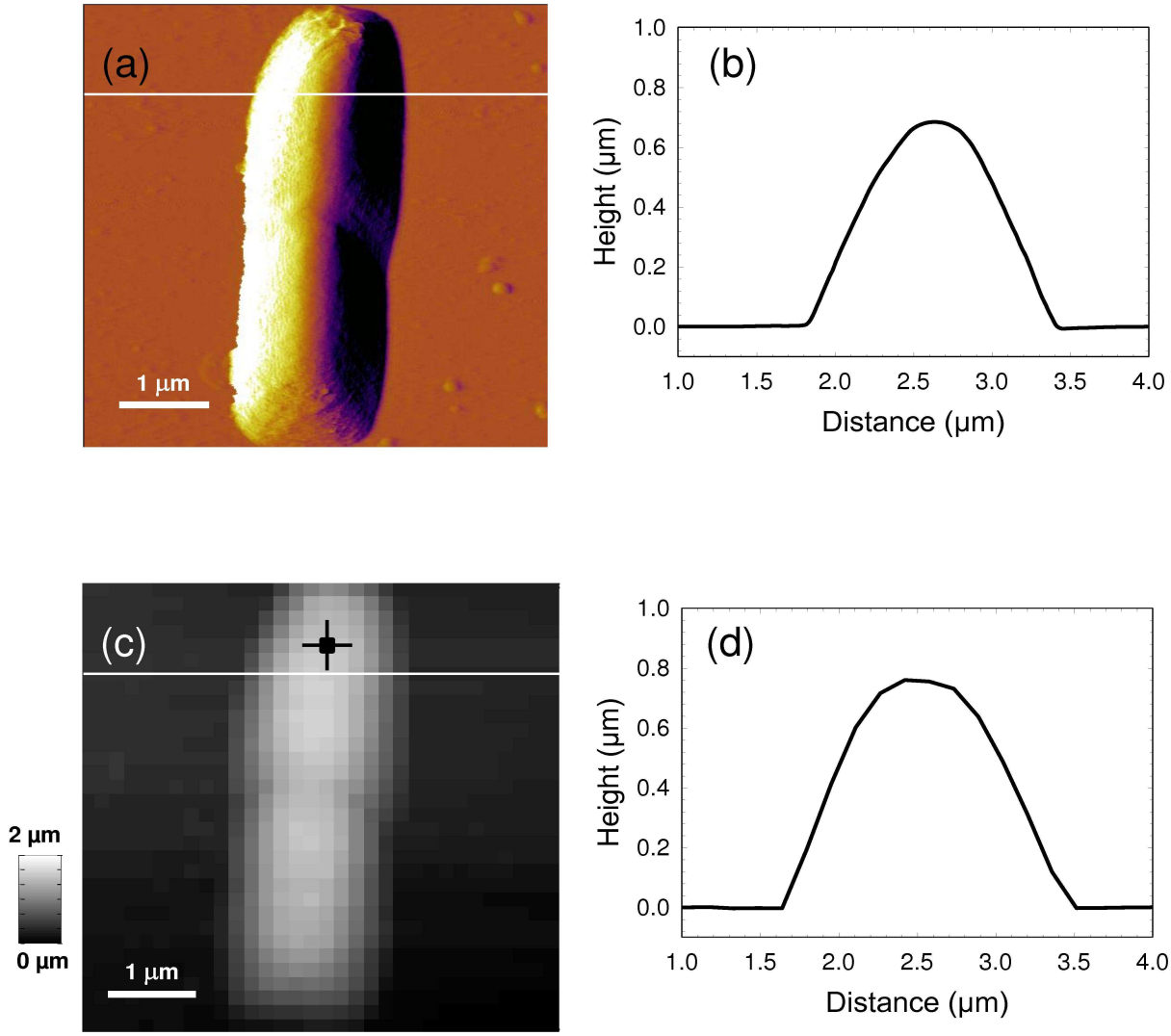


FIGURE 4.4 – Deflection image (a) and reconstructed height image (c) of *E. coli* cells in buffer solution (1 mM KNO_3). Fig. 4.4b and 4.4d correspond to the height profiles measured along the cross section of Fig. 4.4a and Fig. 4.4c as indicated therein by white lines.

mainly attributed to the effects of convolution between tip geometry and cylindrical shape of the bacteria when the latter is subjected to a normal force applied at the edge. Stated differently, the geometry of the contact zone bacterium/tip slightly differs according to whether force is applied in the center or on the edge of the cell. In order to minimize the effects of curvature of cell surface and increase the statistics of the obtained results, we performed additional force volume measurements within a grid of 16-by-16 points on the bacterial surface. The corresponding 500-by-500 nm^2 area is marked by small red squares indicated in Fig. 5a, 5c, 5e and 5g.

Fig. 4.5b and 4.5d show the distributions of the parameters A and κ^{-1} characterizing the electrostatic properties of the bacterial envelope as probed within the aforementioned red square region marked in Fig. 4.5a, 4.5c, 4.5e and 4.5g. In particular, we note an excellent agreement of the most frequent value found for κ^{-1} (11.2 nm) with that expected from theoretical (4.3) which yields $\kappa^{-1}=$ 9.8 nm for a 1mM KNO_3 electrolyte concentration as adopted in the experiment. The Young modulus was found to be 953 $\pm$ 168 kPa (Fig. 4.5f), in qualitative agreement with values obtained by AFM on yeast cells [146] (600 $\pm$ 400 kPa), *Phaeodactylum tricornutum* morphotypes [77] (from $\sim$ 100 to $\sim$ 500 kPa), *Lactobacillus rhamnosus* (186 $\pm$ 40 for wild type and 300 $\pm$ 63 kPa for mutant), *Myxococcus Xanthus* (250 $\pm$ 180 for wild type and 1340 $\pm$ 660 for mutant) in MilliQ water. The value obtained for the bacterial spring constant (0.118 $\pm$ 0.024 N/m in Fig. 4.5h) is in line with that estimated for *Staphylococcus epidermidis* in water [147] (0.24 $\pm$ 0.01 N/m) and for filamentous fungal *hyphae* in PBS buffer [148] (0.29 - 0.17 N/m depending on osmolarity). Schar-Zammaretti *et al.* [149] reported smaller values : 0.05 N/m for *L. crispatus* and 0.03 N/m for *L. helveticus* in 10 mM KH_2PO_4 buffer at pH 7. Finally, we indicate that the results obtained here are very consistent with those reported elsewhere for the same bacteria and obtained according to a manual data processing [99].

4.4.4 Data processing for the retraction force curve

The elastic and mechanical properties of microbial surface macromolecules of *P. fluorescens* as well as their spatial distribution on the surface were probed *via* single-molecule force spectroscopy (SMFS). The 5μm $\times$ 5μm scan region was divided into 32-by-32 pixels similar to the analysis of the approach force curves previously detailed. For each pixel, the modified tip was brought into contact with the biopolymer molecules located at the bacterial surface. The force curves were then recorded upon retraction of the probe tip from the surface of the sample. Fig. 4.3d shows typical data for the force *versus* sample to tip distance together with theoretical reconstruction based on FJC model (4.6). Let us mention that the first 100 nm for each retraction curves are not processed because this part includes ill-defined dependence of the force versus separation curve that possible correspond to the concomitant occurrence of intricate physical phenomena like binding/removal of certain macromolecules on/from the tip or macromolecule rearrangement on the tip surface.

In the segmentation algorithm, both stopping conditions detailed in the 'Algorithm section' are combined. Fixing a maximal value for the number of discontinuity points ensures that the algorithm runs within a tractable computation time (30 seconds at most) while setting a threshold value for the mean approximation error E_{min} is required because the number of regions of interest may strongly differ from one force curve (corresponding to a given pixel in the force-volume image) to another. The E_{min} value is taken identical for all curves. We set $r = 1$ and E_{min} to the empirical variance of the noise estimated from the flat part of the force curve. The segmentation of the retraction curve is illustrated on Fig. 4.2d. Similar to preprocessing of the approach curve, an affine baseline is subtracted from the raw data. Here, the maximal number (20) of discontinuities is reached. For other force curves constituting the force-volume image, the stopping condition relying on E_{min} criterion is met and less than 20 discontinuities are detected.

Fig. 4.6a and 4.6b show the frequency distribution (over the whole FVI) of two fitting parameters involved in the FJC model *e.g.,* the contour length L_c and the Kuhn length l_k. The total contour

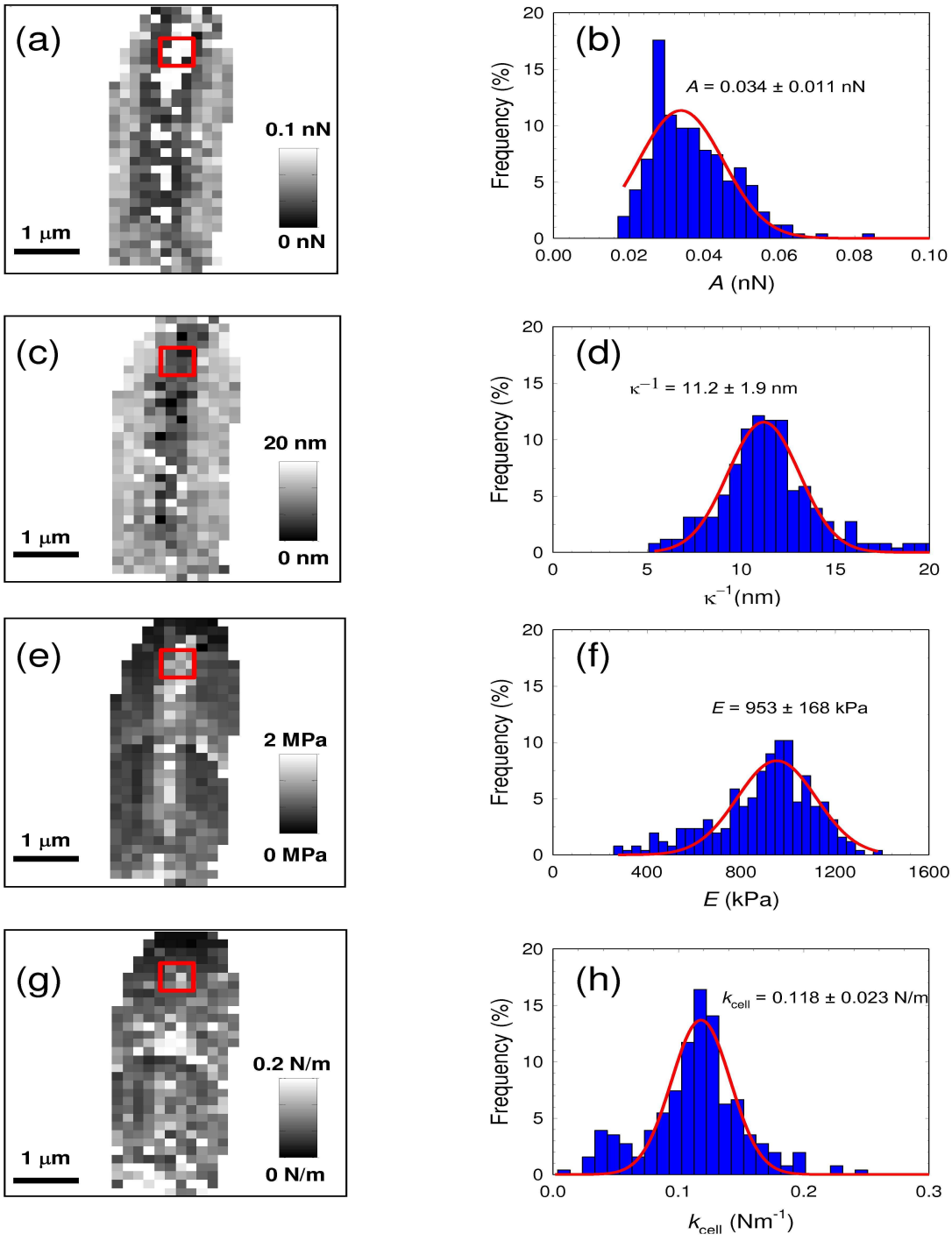


FIGURE 4.5 – Physico-chemical properties of *E. coli* cells in buffer solution (1 mM KNO_3) : (a) 2D map of electrostatic prefactor A (A-range=0-0.1 nN). (b) Statistic distribution of electrostatic prefactor corresponding to 2D map in (a). (c) 2D map of the Debye length (κ^{-1}-range = 0-20 nm). (d) Statistic distribution of the Debye length corresponding to 2D map in (c). (e) 2D map of the Young modulus (E-range = 0-2 MPa). (f) Statistic distribution of the Young modulus corresponding to 2D map in (e). (g) 2D map of the bacterial spring constant (k_{cell}-range = 0-0.2 N/m). (h) Statistic distribution of the bacterial spring constant corresponding to 2D map in (g).

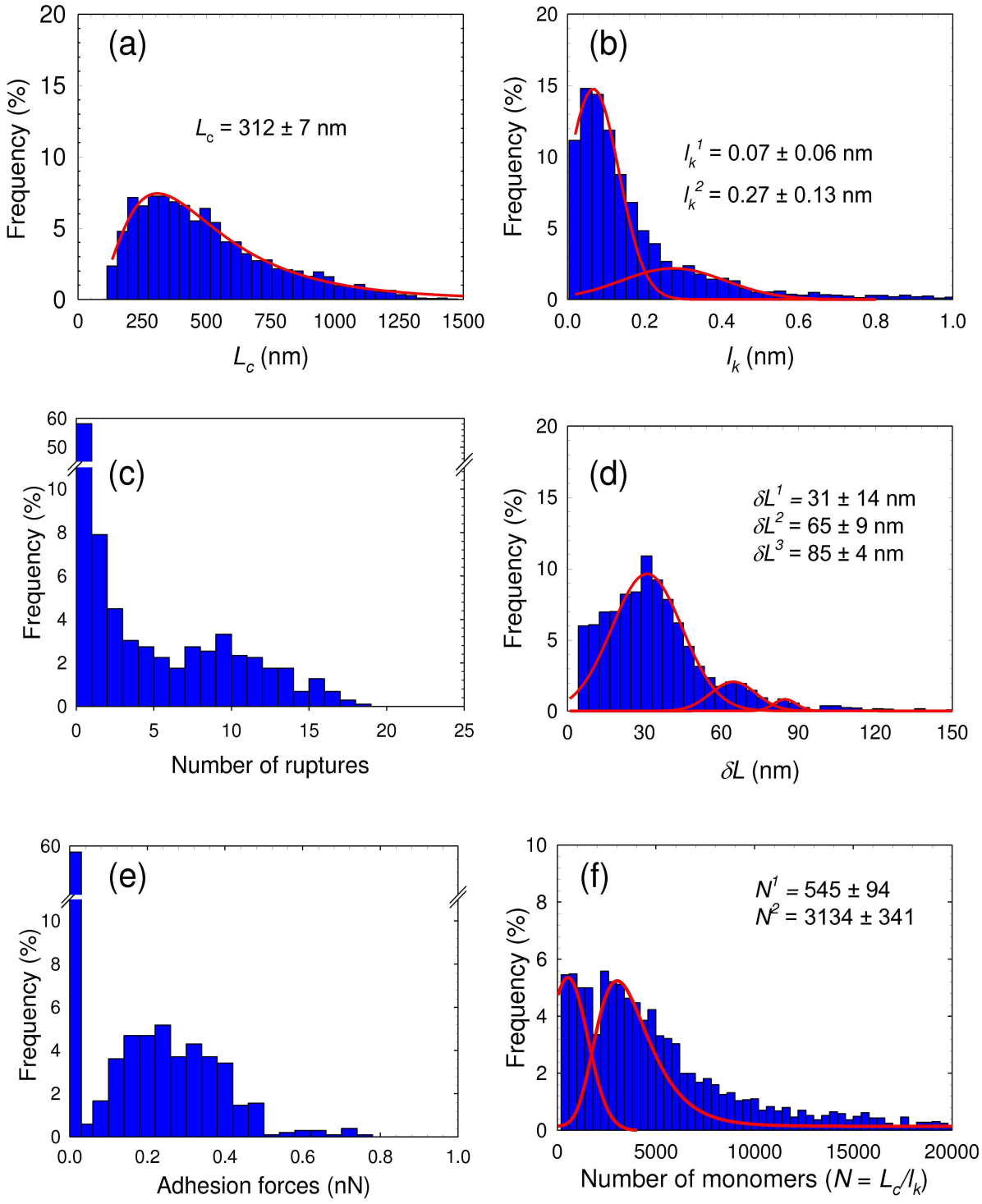


FIGURE 4.6 – Physico-chemical properties of *P. fluorescens exopolymers* : (a) Statistic distribution of the contour length. (b) Statistic distribution of the Kuhn length. (c) Statistic distribution of the number of ruptures (number of regions of interest +1). (d) Statistic distribution of the distance between two consecutive adhesion force ruptures. (e) Statistic distribution of the adhesion force (amplitude of the last adhesive event). (f) Statistic distribution of the number of monomers constituting the polysaccharidic chains ($N = L_c/l_k$).

length of the macromolecule found at the bacterial surface ranges from 100 nm to 1250 nm with a most frequent value of L_c ≈312 nm. The histogram for the distribution of the Kuhn length l_k shows two pronounced maxima at l_k ≈0.07 ± 0.06 nm and 0.27 ± 0.13 nm. The first value is very close to 0.065 nm, which corresponds to the elongation of a single sugar molecule following a conformational change of a C_5-C_6 bond position. The second value is in quantitative agreement with the result of Camesano *et al.* [127] (0.23 ± 0.11 nm for polysaccharides on the surface of *P. putida* in 0.1 M KCl). Furthermore, previous studies evidenced a Kuhn length of about 1.2 nm and 1.4 nm for *Lactobacillus rhamnosus* EPS's [78] which consist in the repetition of 6 sugars [150, 151]. This latter result again indicates that the Kuhn length 0.27 nm we obtain here for *P. fluorescens*, probably corresponds to the size of a glucose molecule. This is confirmed by independent FTIR measurements (data not shown) which demonstrate that *P. fluorescens* produces poly glycogene, strictly composed of glucose units (paper in preparation). Fig. 4.6c shows that from one to twenty uncoiling/rupture event can be detected over the whole retraction curves forming the FVI. The z-distance between two consecutive adhesion events δL (see Fig. 4.1d for the definition of δL) is related to the length between two branches in the poly glycogen chain. The histogram representing the frequency distribution of δL over the whole FVI (Fig. 4.6d) yields three pronounced maxima at 31 ± 14 nm, 65 ± 9 nm and 85 ± 4 nm. The observed periodical length l_p of ∼ 30 nm can be attributed to the shortest distance between two branches within a given poly glycogen chain, since unfolding may occur simultaneously on the various parts of the macromolecule. In addition, the histogram (Fig. 4.6f) which depicts the distribution in the number of monomers distribution ($N = L_c/l_k$), exhibits two peaks. The first peak accounts for about 25% of the polysaccharidic chains with 545 monomers, while the second peak corresponds to about 70% of the polysaccharidic chains with 3134 monomers. This result suggests the presence of both short and very long polysaccharidic chains, which is in line with previous studies carried out on the ramified structure of glycogen (see Fig. 4.7d). Melédez-Hevia *et al.* [152, 153] optimized several structural parameters of glycogen to achieve an efficient fuel storage molecule in the cell. Their results show that this molecule is formed by concentric poly glycogene fractal chains with 12 branches (see Fig. 4.7d). A rough estimation from our data suggests $L_c/l_p \approx 10$ branches. The discrepancy may originate from the fact that the fuel storage capacity is not critically governed by the structure of poly glycogene outside the cell. The statistical distribution of adhesion forces, shown in Fig. 4.6e, evidenced that polysaccharides are absent or not detectable from more than 60% of the total FVI area. Furthermore, we attribute the adhesion forces in the range 0.1 - 0.5 nN to the simultaneous detection of two to ten macromolecules, as judged from independent calibration SMFS measurements performed on glucosamine grafted-gold surface (see Supplementary Materials) and litterature [138, 154]. The distribution of adhesion force is located around the bacterial cells (Fig. 4.7a and 4.7b), thereby suggesting that *P. fluorescens* excretes EPS outside the cell wall.

Fig. 4.7a shows the deflection image obtained in contact mode for *P. fluorescens*. In order to quantitatively clarify the distribution of the number of macromolecules on the surface, we considered the adhesion force corresponding to the last peak detected in the retraction curves. The amplitude of the adhesion force in this case is proportional to the number of macromolecules simultaneously detached from the tip. The corresponding 2D map is presented in Fig. 4.7b. To better visualize differences in polysaccharide properties, three dimensional map was constructed by combining last rupture distances (z level) and adhesion forces (colors) measured at every (x, y) location (see Fig. 4.7c). From direct comparison of Fig. 4.7a, 4.7b and 4.7c, it turns out that the cell is not covered entirely by polysaccharides which are distributed mostly around the cell. To conclude this part, we mention that not only the results are in good agreement with those of the AFM literature but they are also confirmed by other technique such as FTIR. In addition, to the best of our knowledge, these results provide the first complete large scale physico-chemical analysis (*i.e.,* on/outside the bacterial cell) of such macromolecules on living organisms.

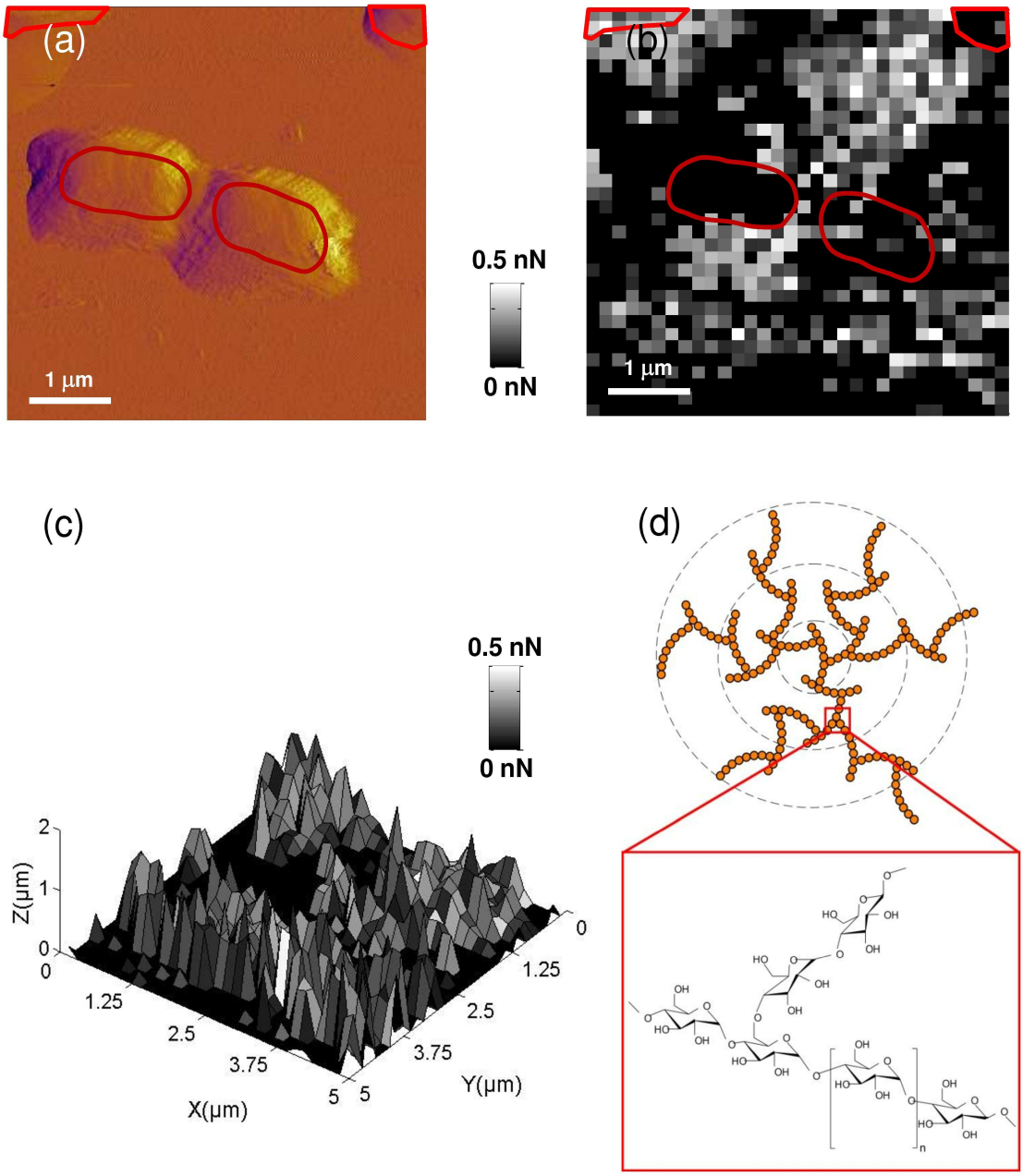


FIGURE 4.7 – Maps of adhesive properties of *P. fluorescens*, and glycogene structure : (a) Deflection image of *Pseudomonas fluorescens* taken before FVI experiment, the red zones correspond to bacterial cells location. (b) Adhesion force map (F-range = 0-0.5 nN) for the last adhesion force rupture measured on bacteria located on the surface as indicated in panel a. (c) Three dimensional map of adhesive properties of *P. fluorescens* combining the last adhesion force and last rupture distance corresponding to uncoiled exopolymers. (d) Fractal structure of the poly glycogene and chemical formula.

4.4.5 Discussion on the performance of the algorithm for force curve analysis

While in the previous parts, the focus was mainly given to the relevance of the estimated physical parameter values, in the following we address the evaluation of the proposed algorithm from a signal processing perspective. We also discuss some possible extensions of the algorithm.

The force curve segmentation algorithm can be seen as a *signal compression* tool. Indeed, a force curve originally contains thousands of data points, and the segmentation algorithm performs a piecewise-smooth approximation based on a very limited number of discontinuity points (less than 50). The piecewise-smooth approximation can be interpreted as a compressed version of the original signal since it can be coded based on a few values (the number of discontinuity points) in comparison with the total number of data. Clearly, the length of coding depends on the stopping conditions of the segmentation algorithm, *e.g.,* the maximal number of iterations and/or the threshold value for E_{min} pertaining to the mean approximation error. When the number of discontinuity points increases (or when E_{min} is set to a low value), the approximation error is reduced, but on the other hand, the number of regions of interest (related to the concept of coding length in information theory) increases. When the number of discontinuity points is large, the main signal features (main slopes, main curvatures) are embedded in the approximated signal. The error signal, defined as the difference between the observation and the piecewise smooth approximation signals, is mainly composed of observation noise. When dealing with a force-volume image (typically, about 32 $\times$ 32 $\times$ 3000 data points, *i.e.,* more than 10^6 signal values must be stored per force volume image), it is of interest to store the approximated force curves on a computer, since this storage can be done with a drastic save of memory.

Let us now discuss the behavior and the potential extensions of the force curve algorithm presented above. We discuss possible improvement for increasing the speed of the algorithm and adaptation to physical models other than those presented here. Among the three steps of the algorithm, the segmentation procedure is the most time consuming, depending on the number of detected discontinuities (*e.g.,* number of iterations) and on the degree r for the piecewise polynomial. For approach curves, only 6 iterations are performed, and the segmentation algorithm then typically runs over less than 2 seconds. For retraction curves, the number of iterations can reach 40 or 50 when analyzing very long *Worm Like Chains* or *Freely Jointed Chains*. For such situations, the segmentation algorithm runs within 30 seconds to 1 minute. We developed the algorithms in Matlab. These time scales are only informative, and could obviously be reduced with an implementation of the algorithm in a compiled language (*e.g.,* in C). The two other procedures (detection of the regions of interest and parametric model fitting) are very fast. The detection of the regions of interest from the segmentation results is almost instantaneous, and the parameter estimation (local optimization algorithm) runs within a few seconds. The latter step can reach up to 5 seconds in the retraction case depending on the number of intervals where local optimization is performed.

Although the algorithm has been described for specific parametric models (electrostatic interaction, Hertz and Hooke interaction, FJC model), it should be recognized that the fitting step can be extended to other models. In the approach case, other Hertz models could be used within the same optimization framework. For instance, when the shape of the AFM tip is modeled as a sphere, the Hertz model differs and reads $F_{Hertz}(\delta) = B\delta^{3/2}$ instead of $F_{Hertz}(\delta) = B\delta^2$ [117]. A more precise expression was proposed by Dimitriadis *et al.* [97]. An extension of the present work could easily include these more complex models in the fitting algorithm.

In the retraction case, several adaptations of the proposed algorithm are possible. First, let us notice that the fitting of experimental data to the FJC model in each region of interest $\mathcal{R}_j$ leads to the estimation of the persistence length l_k^j and the contour length $L_c^j = L_c(l_k^j)$, the number of monomers reading as $N_j = L_c^j/l_k^j$. Clearly, this quantity is generally not an integer. If one desires to impose the constrain that this number is an integer, one needs to perform a small post-processing for replacing N_j by its the closest integer, and subsequently updating the values of l_k^j and $L_c^j = N_j l_k^j$. This post-processing takes the form of a 1D local optimization problem. We observed that this

post-processing leads to very minor changes of l_k^j and L_c^j values and, to a negligible increase in the approximation error.

Similarly to the approach case, we considered the adaptation of the algorithm to other retraction models. We performed the fitting of data according to the WLC model [155, 156, 157] by designing an appropriate least-square cost function. Contrary to the cost function defined in (10), the squared error now relies on the difference between the observed F_i values and their approximation based on the WLC model which takes the form F_{WLC}. The WLC parameter estimation can be performed successfully under appropriate specification of mild constraints on the parameter values and realistic setting of their initial guesses. This kind of model is usually adapted to protein and polypeptides.

For polysaccharide macromolecules, the generally accepted model is FJC+ which takes into account monomer elasticity. Using the FJC+ model rather than FJC enables us to improve the quality of fit, *i.e.,* to decrease the error between the experimental data and their approximation in each region of interest. This improvement is related to the additional parameter k_s involved in the FJC+ model (4.7). We developed an algorithm that is adapted to this kind of modeling. However, when subsequently processing the curves of a given FVI, we a observed two kinds of behavior. In the first case, the obtained FJC+ parameters are very realistic and in coherence with their expected values. On the contrary, numerical issues occur for a number of pixels where the cost function to be minimized (corresponding to the approximation error in the given region of interest) takes a degenerate form. Indeed the minimizer of the cost function may be located inside a very flat valley, making the fitting problem very ill-conditioned. In other words, a large range of values for l_k, L_c and k_s leads to the same approximation error. In such case, we verified that it is mandatory to impose additional constraints (*e.g.,* forbidding some ranges of values for l_k, L_c or k_s) to cope with the numerical optimization issues and obtain reliable and realistic estimations of the relevant physical parameters. It is beyond the scope of the current study to discuss these features in details.

4.5 Conclusion

We proposed two original automatic algorithms for processing approach and retraction AFM force curves. Their common feature is the accurate detection of key regions of interest where available force models may be applied for quantitative interpretation of the data. We automatically processed several force-volume images by running the algorithms for analyzing all force curves recorded at the pixels constituting the image. This processing allows us to provide statistical and spatial distributions of electrostatic and nanomechanical properties of biological samples together with characteristic features pertaining to their outer polymeric materials (specific adhesion). An interesting perspective of the work consists in considering several force curves jointly while the present algorithm processes each force curve in an independent manner. It is indeed of interest to take into account the fact that some of the physical parameters (*e.g.,* the topography of the sample) do not strongly vary from a given pixel to its neighboring pixels. An extended tool would then estimate physical parameters relative to a pixel based not only on the corresponding force curve but also on the force curves recorded at the neighboring pixels. This would allow for finer parameter estimation and faster data processing.

The proposed algorithm was tested on real data obtained with AFM. The approach experiments were performed on *E. coli* bacterium without surface appendage. For the retraction case, we investigated surface properties of *P. fluorescens*, which produces exopolysaccharides (EPS). For the approach force curves, the detected discontinuities are associated to critical points marking the transition between regions where electrostatic, non-linear (Hertz) and linear (Hooke) contact mechanical interactions take place. For retraction curves, the detected points are related either to unfolding or to detachment of the EPS macromolecules from the AFM tip.

The data are then fitted to theoretical expressions for obtaining relevant physical parameters pertaining to *E. coli* bacterial cell wall : the Debye length, Young modulus (stiffness of bacterial envelope) and turgor pressure (inner osmotic pressure). In addition we analyzed detailed structural, conformational and mechanical properties of exopolysaccharides from *P. fluorescens* cell wall, *i.e.,* Kuhn length, contour length and parameters describing the the fractal structure of the macromolecule [153, 158, 159]. Inspection of the obtained spatial distribution clearly shows an accumulation of polysaccharides around *P. fluorescens* and not only over the cell wall. The results are in quantitative agreement with data from literature. Several benefits in using the proposed algorithm need to be stressed. For the first time, electrostatic properties of bacterial envelope without surface appendage are extracted. Detection of the transition points between different force regimes increases the reliability of the obtained results for the approach experiments and makes it possible to perform automatic fitting of retraction curves without manual selection of the regions of interests. An extension of the present work will include more complex and precise models developed for elastic deformation [97, 117]. Another interesting issue is the characterization of animal cell biomacromolecules and calculation of polymer layer density (on eukaryotic cells) by molecular stretching and detection of polymer steric forces.

Chapitre 5

Separate sparse sources from a large number of shifted mixtures

5.1 Introduction and motivation

As was introduced in Chapter 1, at each pixel (x_i, y_i) on the 2D grid $\{(x_i, y_i)|i = 1, 2, \ldots, M\}$ representing the sample surface, the AFM measures a 1D force curve $f_i(z) = f(x_i, y_i, z)$ recording the evolution of the probe-sample force f w.r.t. the piezo displacement z. Measurements for all M pixels yield M force curves, namely, a 3D data set $f(x, y, z)$, which is called force-volume image. As a notable advantage of AFM, the resolution in z can be as precise as nanometer, yielding thousands of samples in a force curve. When the 2D grid reaches *e.g.,* $M = 256 \times 256$ pixels, 3D force-volume images require a large memory storage. To process such image, one need to develop algorithms with an efficient implementation.

In Chapter 2, we developed a sparse approximation algorithm (SBR), its continuation version (CSBR) is developed in Chapter 3, and applied these algorithms to approximate a force curve by using the piecewise polynomial dictionary, in which each elementary signal contains a discontinuity. The sparse representation of a force curve then relies on the locations, amplitudes and orders of its discontinuities, and the approximation quality is controlled by the number of discontinuities, *i.e.,* the number of selected elementary signals. By **sparsifying**, we can represent a force curve by a sparse spike train, *i.e.,* the sequence of the detected discontinuities, therefore drastically decreasing the complexity of the signal. This is close to the idea of lossy compression, where an hyperparameter handles the tradeoff between encoding length and loss.

Once the discontinuities have been estimated, we can fit each piece of a force curve using the available physical models like Hertz model, worm-like chain or freely jointed chain [2]. Informative physical quantities like Young's modulus or contour length can be estimated from each approximated force curve, and then the collection of the parameters estimated for all pixels can be displayed in a 2D *x vs y* image or a 3D *x vs y vs z* image, which represents the spatial distribution of the physical quantities. Such an approach was presented in Chapter 4.

However, such an approach suffer from two main shortcomings :

- it mainly consists in processing the M force curves separately. By a joint processing of the force curves, it is expected to decompose the data into a limited number of region, each region being characterized by a different force curve ;
- it supposes that each force curve can be decomposed into a finite (limited) number of physical interactions, thus requiring to fix, *a priori*, different kinds of interactions. By performing a blind separation, it is expected to decompose the force volume image into regions sharing the

same behaviors, from which the different type of physical interactions can be inferred.

In [5], mixture model of force volume image has been proposed and validated on real data. Each approach force curve can be modeled as linear combination of delayed elementary interactions, yielding a mixing model of shifted sources, also referred to as anechoic mixture in the acoustic source separation literature [160]. However, before going further, we have to mention that contrary to the case of acoustic source separation, where generally only a limited number of mixture are available, in force volume image unmixing, a large number of mixtures are available. In that respect, force volume image unmixing is similar to hyperspectral image. However, the mixture models being different in the two cases, there is a need for developing methods allowing to unmix a large number of shifted mixtures.

This chapter is organized as follows : we first introduce the sparse source separation problem from shifted mixtures, followed by a brief bibliography. Secondly, modeling, ambiguities and conditions for identifiability are discussed. Thirdly, an algorithm to separate shifted mixtures of sparse spike trains is developed for the noise-free setting, and then extended to noisy setting. And finally the proposed algorithm is tested on both simulated and real data.

5.2 Sparse source separation problem

Source separation is an old problem, originating from the so called "cocktail party problems" [161], where one wants to distinguish P speakers from M observed acoustic signals. Each speaker's voice is referred to as a source, and each observed acoustic signal is referred to as a mixture. Source separation problems also occur in molecular spectroscopy. In Raman and Infrared (IR) spectroscopy [162], measured spectra (mixtures) depend on a frequency or wavelength dimension. Pure spectra (sources) are needed to identify the sample constituents and mixing weights are needed to assess their concentrations in the measured spectra. In source localization, *i.e.,* global positioning system (GPS), in order to focus on a source signal among the interferences and background noise, one wants to localize a single source signal by using source separation techniques.

In AFM force volume image, we consider a force curve for a pixel composed of several elementary interactions representing different components. *e.g.,* goethite sample lies on a glass slide, the measured force curves for the border pixels consist the influences from both goethite sample and glass slide, while the measured force curves for the pixels within goethite sample only consist the influence from the goethite sample. Therefore the measured force curve for each pixel can be modeled as an anechoic mixture of elementary interactions [5]. Extracting of elementary interactions can help the physicist better understand the interaction between the probe and specimen. Furthermore, once the elementary interaction are estimated, we can estimate the spatial shift of each elementary interaction in the measured force curve for each pixel, yielding the 2D topography of each elementary interaction, which is informative in surface science.

5.2.1 Different kinds of mixing models

In the speech signal processing literature, the mixing models are classified into three classes w.r.t. the complexity of the propagating environment [160] (see Tab. 5.1) : *instantaneous mixing, anechoic mixing* and *echoic mixing.*

1. Instantaneous mixing only considers the attenuations of the propagation without delays (shift in spatial domain), that is to say, the propagating path from p-th source to m-th sensor is modeled as scalar multiplication with weight $c_{p,m}$.
   The instantaneous mixing is the most common used model in applications, thus algorithms for instantaneous unmixing are fruitful. Popular methods are : Independent Component Analysis (ICA) [163, 164], Joint Approximate Diagonalization of Eigen-matrices (JADE) [165],

TABLE 5.1 – Mixing model specific linear operator and mixing parameters [160]

Mixing model	Linear operator	Generative model	Entry of $\boldsymbol{C}$
Instantaneous	Matrix multiply	$\boldsymbol{f}(t) = \boldsymbol{C}\boldsymbol{s}(t)$	$c_{p,m}$
Anechoic	Convolution	$\boldsymbol{f}(t) = \boldsymbol{C} * \boldsymbol{s}(t)$	$c_{p,m}\delta(t - \tau_{p,m})$
Echoic	Convolution	$\boldsymbol{f}(t) = \boldsymbol{C} * \boldsymbol{s}(t)$	$\sum_{l=1}^{K_{p,m}} c_{p,m}^{l}\delta(t - \tau_{p,m}^{l})$

BS-InfoMax [166]. The main assumption is : the sources are statistically independent, which is supposed to be true in acoustic signal processing. In recent decades, sparse assumption has received intensive study [160, 167, 168]. We say that a signal is sparse when most of its coefficients are zeros in some transform domain. Speech signal can also be assumed to be sparse in time-frequency domain [169]. In hyperspectrum imaging, sources are assumed to be non-negative, yielding the flourish of Non-negative Matrix Factorization (NMF), which is source separation problem under the assumption than both sources and mixing process are non-negative. Interestingly, NMF always leads to spares representation [170], which inspired Non-negative Sparse Coding (NSC) [171].

2. Anechoic mixing considers the direct propagation from sources to sensors with both attenuations and delays (shift in spatial domain), *anechoic* means there is no reflection which is caused by obstacles like walls. So the propagating path from p-th source to m-th mixture is modeled as a $c_{p,m}$ weighted, $\tau_{p,m}$ delayed impulse function $c_{p,m}\delta(t - \tau_{p,m})$, here $\delta(t)$ is the Dirac symbol.

   In [172], Torkkola extended BS-InfoMax to anechoic unmixing. In [18], Bofill also extended the idea in [173] to anechoic unmixing by using Fourier transform as the preprocessing, then matrix of relative attenuations and matrix of delays can be inferred by using clustering technique. Degenerate Unmixing Estimation Technique (DUET) is a most remarkable method to address anechoic mixing of speech signal in the under-determined case. The basic assumption of DUET is : the sources are *W-disjoint orthogonal* [174] : there is no overlap among the supports of all the sources in the time-frequency domain. In fact, W-disjoint orthogonal is a sparsity assumption. We will discuss DUET later in detail.

3. Echoic mixing is the most general model, where the reflected signals are also taken into account. So the propagating path is modeled as a Finite Impulse Response (FIR) Filter with several delays. This model depicts the complex propagating environments, *i.e.,* the pathway between a cell phone and a base station. Lambert study the unmixing of echoic mixing in his thesis [175].

Both anechoic and echoic mixing are also referred to as convolutive mixing, which is studied intensively in the survey paper [176]. In [177], Jafari considered the convolutive blind source separation problem of stereo mixtures. The proposed Adaptive Stereo Basis (ASB) uses adaptive transform basis which is learned from the stereo mixture pairs. As the author pointed out, ASB does not make any specific assumptions regarding the mixing channel, the learned basis pairs should automatically capture the nature of the channels. As was shown in experiments, the basis learned from speech signals localized well in time and frequency domain, yielding representations that exhibit both wavelet- and Fourier-like basis.

Because the AFM force volume image are modeled as anechoic mixtures [5], we will mainly focus on anechoic unmixing. The anechoic mixing often appears in speech signal processing, where usually one has less mixtures than sources ($M < P$). This problem is referred to as degenerate or underdetermined unmixing. However, in AFM force volume image, we have much more mixtures than sources ($M \gg P$, typically we want to estimate several sources from thousands of mixtures), so most algorithms cannot be used to our problem directly.

AFM force volume image shares the similarity with hyperspectral image in the sense of overdetermined unmixing ($M > P$). There exist many methods to unmix hyperspectral image [178, 179],

however because of the different mixing models (hyperspectral image is instantaneous mixture while AFM force volume image is anechoic mixture), the methods for hyperspectral image cannot be employed to solve our problem. Up to our knowledge, there does not exist algorithm for our problem, either because of the large number of mixtures, or the anechoic mixing model. So a new method needs to be developed for our problem.

5.2.2 Separation of instantaneous mixture of sparse sources

In order to help the reader easier understand the proposed algorithm, in this subsection, we introduce the main idea of instantaneous unmixing based on sparsity assumption. In the next subsection, we introduce the main idea of DUET for anechoic unmixing, which inspire the proposed algorithm. Afterwards, we introduce a common structure of sparse source separation methods, and analysis the similarity and difference between the proposed algorithm and DUET within the common structure.

For instantaneous mixture, after sparsifying we have :

$$\boldsymbol{f} = \boldsymbol{C}\boldsymbol{s} \tag{5.1}$$

where $\boldsymbol{f}$ and $\boldsymbol{s}$ are the sparse coefficients of data point of observations and sources respectively. $\boldsymbol{C}$ is the mixing matrix. Decomposing $\boldsymbol{f}$ along the columns of $\boldsymbol{C}$ we have

$$\boldsymbol{f} = \boldsymbol{c}_1 s_1 + \ldots + \boldsymbol{c}_P s_P \tag{5.2}$$

where $\boldsymbol{c}_p$ is the p-th column of $\boldsymbol{C}$ and s_p is the p-th entry of $\boldsymbol{s}$. When we assume $\boldsymbol{s}$ is a sparse vector, *i.e.,* only a particular source i is significantly different from zero, for data point $\boldsymbol{f}$, we obtain :

$$\boldsymbol{f} = \boldsymbol{c}_i s_i + \epsilon \tag{5.3}$$

where ϵ is the perturbation due to the remaining sources. We can see data point $\boldsymbol{f}$ is proportional to $\boldsymbol{c}_i$ with scale factor s_i if ϵ is small enough. So under the assumption of sparsity, most data points with significant length belongs to only one source, thus data points belonging to i-th source will cluster along the direction defined by $\boldsymbol{c}_i$. Each column of $\boldsymbol{C}$ can then be estimated from the scatter plot of data points (see Fig. 5.1). It is worth noticing that because neither $\boldsymbol{c}_i$ nor s_i is known, we can only estimate the relative length of $\boldsymbol{c}_i$.

5.2.3 Degenerate Unmixing Estimation Technique (DUET)

The proposed algorithm in this chapter is inspired by DUET, so we give a brief introduction of DUET here. In Degenerate Unmixing Estimation Technique (DUET) [174], Yilmaz modeled the two mixtures as :

$$f_1(t) = \sum_{p=1}^{P} s_p(t) \tag{5.4}$$

$$f_2(t) = \sum_{p=1}^{P} c_p s_p(t - \tau_p) \tag{5.5}$$

where $s_p(t)$, c_p and τ_p are the p-th source, weight and delay. We use Δ to denote the maximal possible delay between sensors, and thus, $|\tau_p| \leq \Delta, \forall p$. Give a windowing function $W(t)$, the

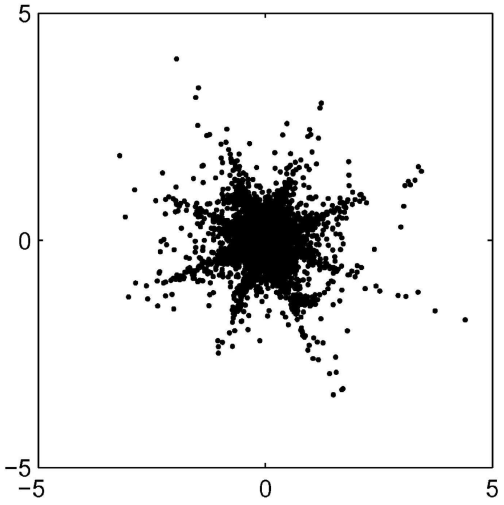


FIGURE 5.1 – Two-sensor scatter plot of the instantaneous mixture of four voices. Figure comes from [18]

windowed Fourier transforms of $s_p(t)$ is defined

$$\mathcal{F}^W(s_p(\cdot))(\omega,\gamma) = \int_{-\infty}^{+\infty} W(t-\gamma)s_p(t)e^{-j\omega t}dt$$

which we will refer to as $S_p^W(\omega,\gamma)$. In the time-frequency domain, (5.4) and (5.5) reread :

$$\begin{bmatrix} F_1^W(\omega,\gamma) \\ F_2^W(\omega,\gamma) \end{bmatrix} = \begin{bmatrix} 1 & \dots & 1 \\ c_1 e^{-j\omega\tau_1} & \dots & c_P e^{-j\omega\tau_P} \end{bmatrix} \begin{bmatrix} S_1^W(\omega,\gamma) \\ \vdots \\ S_P^W(\omega,\gamma) \end{bmatrix} \tag{5.6}$$

When $S_p^W(\omega,\gamma)$ is assumed as *W-disjoint orthogonal* : the supports of the windowed Fourier transform of $s_{p_1}(t)$ and $s_{p_2}(t)$ are disjoint, then at most one of the P sources will be non-zero for a given (ω,γ), thus

$$\begin{bmatrix} F_1^W(\omega,\gamma) \\ F_2^W(\omega,\gamma) \end{bmatrix} = \begin{bmatrix} 1 \\ c_i e^{-j\omega\tau_i} \end{bmatrix} S_i^W(\omega,\gamma) \text{ for some } i \tag{5.7}$$

therefore, we can calculate the relative amplitude and delay parameters associated with one source using

$$(\hat{c}_i,\hat{\tau}_i) = \left(\left\| \frac{F_2^W(\omega,\gamma)}{F_1^W(\omega,\gamma)} \right\|, \Im(log(\frac{F_1^W(\omega,\gamma)}{F_2^W(\omega,\gamma)}))/\omega \right) \tag{5.8}$$

$\Im$ denotes taking the imaginary part. We can see at each time-frequency point, from $(F_1^W(\omega,\gamma), F_2^W(\omega,\gamma))$, pair $(\hat{c}_i,\hat{\tau}_i)$ can be calculated following (5.8). When we draw the pairs at all time-frequency points on a 2D attenuation c *vs* delay τ histogram, $(\hat{c}_i,\hat{\tau}_i)$ can be accumulated, yielding P peaks in the histogram, each peak corresponds to one source. See Fig. 5.2.

Once the estimates of mixing parameters $\hat{c}_i$ and $\hat{\tau}_i$ are known, source representations can be estimated via time-frequency masking. For each source, say i-th source, we can construct two time-frequency masks for two mixtures. Each pixel on the mask denotes the contribution (can be binary or fractional) of the i-th source to the mixture at this pixel. The sum of the two masked mixtures

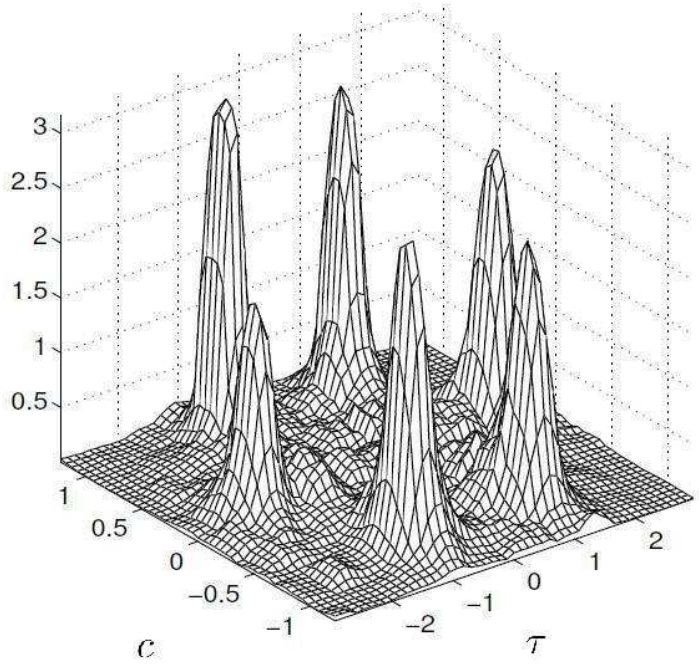


FIGURE 5.2 – Six sources synthetic mixing smoothed histogram. Each peak corresponds to one source and the peak location corresponds to the associated source's mixing parameters. Figure comes from [174]

yields the estimate of the i-th source. The contribution can be calculated from the distance between the pair $(\hat{c}, \hat{\tau})$ at the pixel and the i-th source's parameter estimate $(\hat{c}_i, \hat{\tau}_i)$ on the c *vs* τ plane.

In a word, the two mains enlighting points in DUET is :

- Working in the 2D plane is better than working in the 1D plane, thus we have one more dimension to distinguish sources. From Fig. 5.2, we can see the six sources have separability in the 2D plane.
- Accumulating the delay and attenuation. This is different from most other demixing methods.

These two points will be discussed in detail in the next subsection.

The main shortcoming of DUET is the poor performance for large time delay (larger than one sample). To overcome this, In [180], Cho proposed to find a proper frequency range that produces no phase ambiguity.

5.2.4 Enlighting points of existing algorithms

In this subsection, firstly we introduce a common structure of sparse source separation methods, and then analysis the similarity and difference between the proposed algorithm and DUET.

As Gribonval summarize in [167], sparse source separation methods usually consist of three steps :

1. Firstly a sparsifying linear transform is applied to the mixtures (*e.g.,* Fourier transform) ;
2. Estimating the mixing parameters from the scatter plot of the sparsified mixtures ;
3. Estimating the sparse source representations.

This structure is suitable for underdetermined unmixing where one has less mixtures than sources. Since in underdetermined case, when neither sources nor mixing parameters are known, because we have less mixing parameters to estimate (2 parameters, τ, c in each mixture, totally $2M$ parameters) than parameters in sources (in each frequency bin, 2 parameters, *i.e.,* real and imaginary part, L bins, R time frames, P sources, totally $2LPR$ parameters), estimating the mixing parameters first is

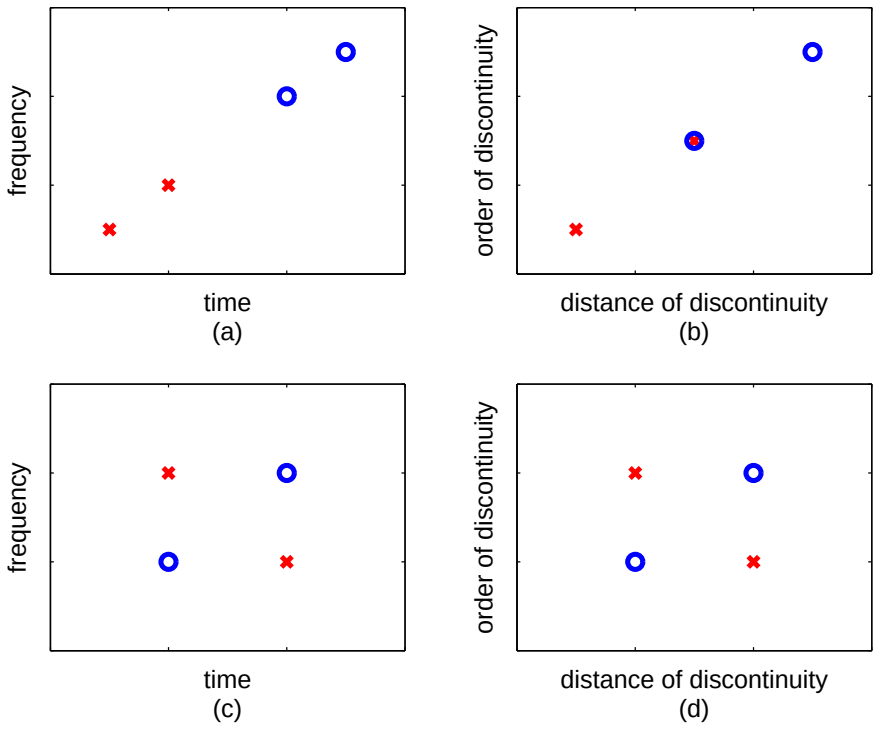


FIGURE 5.3 – Working domain and separability of speech signal and AFM force volume image. (a) and (c) show time-frequency domain for speech signal while (b) and (d) show distance of discontinuity *vs* order of discontinuity domain. In each subfigure, cross (X) represents one source and circle (O) represents the other source. In (a) the two sources can be separated because their support are disjoint in time domain, frequency domain or time-frequency domain. In (c) the two sources can be separated in only time-frequency domain because the supports of their marginal distribution on time or frequency domain are conjoint. (d) is the same case as (c) by just changing the meaning of two coordinate axis. (b) is a worst case, the two sources cannot be separated.

easier than estimating the sources first. On the contrary, when we have less sources than mixtures, estimating the sources first is easier. As a result, in our proposed algorithm, we exchange the order of step 2 and 3. *i.e.,* estimating sparse source representations before estimating the mixing parameters.

In step 1, speech signal is often assumed to be sparse in the time-frequency domain [169]. So STFT is the ideal sparsify linear transform. However AFM data are not supposed to be sparse in time-frequency domain. Nevertheless, in Chapter 2 and 3 we showed that we can sparsify force curves in piecewise polynomial dictionary by using CSBR algorithm. Fig. 5.3 illustrates the working domain and separability of speech signal and AFM force volume image. The exact meaning of distance of discontinuity *vs* order of discontinuity will be explained later. It is worth noticing that the different working domain leads to the different methodology (according to whether the delay still exist or not in the new domain) in the next two steps. Short Time Fourier Transform (STFT) can transform delay in temporal domain into complex multiplication in frequency domain (true under narrowband assumption), anechoic mixtures in temporal domain then become "instantaneous" (not exactly) mixtures in frequency domain thanks to the "time translation" property of Fourier transform. Thus methods for instantaneous unmixing can be adapted. However, our proposed sparsifying transform **keeps the delay in the new domain** because of the using of Toeplitz matrix as sparsifying dictionary, new methodology to unmix anechoic sparse spike trains needs to be developed in the next two steps.

The step 2 of DUET is inspirational. In DUET, the *relative* attenuation $\hat{c}_i$ and *relative* delay $\hat{\tau}_i$ can be accumulated in the 2D c *vs* τ histogram. Inspired by this, we find that in the anechoic mixtures of sparse spike trains, the *relative* location of spikes, *i.e.,* interval of spikes from the same source, can be accumulated in a histogram. The explanation is when the delays of the sources in mixtures vary, the interval of spikes from the same source do not vary, while that from the different sources do vary. So when we analysis all the intervals of lots of mixtures in a histogram, several peaks appear, each corresponds to one interval whose two spikes belonging to the same source. These peaks can be used to estimate sources by linking the spikes in the mixtures that belong to the same source.

Our proposed step 3 is totally different from DUET. In DUET, we use time-frequency masking to estimate source representation. In our proposed algorithm, we match each mixture sparse spike trains with the estimated source sparse spike trains, which leads to discrete searching method. Once the mixture is matched, the estimation of mixing parameters is obvious. In the next section, we start from modeling and identifiability of sparse source separation problem.

In [181], source representations are estimated by minimizing their ℓ^q-norm ($0 < q < 1$) under the constrain that the reconstructed mixtures equal to the observed mixtures. While in [18], ℓ^q-norm is replaced by magnitude.

It is necessary to emphasize the different assumption used in [18], DUET and our proposed method. Because in the blind source separation problem, neither the sources nor the mixing parameters are known. So we could assume one of them as a fixed quantity, and the other one as a random variable. The fixed one can be estimated firstly without knowing the other one, then the realization of the random variable can be estimated afterwards. Both in [18] and DUET, one has limited number of mixtures, so the mixing parameters are assumed fixed, while the coefficients of sources on frequency bins are assumed as random variables, therefore the mixing parameters are estimated firstly. However for our proposed method, because we have thousands of mixtures but several sparse sources, so on the contrary, we assume the mixing parameters as random variables, and the intervals of discontinuities of sources as fixed quantities, and thus be estimated firstly.

- Brief introduction of DUET
  Degenerate Unmixing Estimation Technique (DUET) [174] is one of the most successful method to address anechoic mixing of speech signals in the under-determined case. The basic assumption of DUET is : the sources are *W-disjoint orthogonal*, which means there is no overlap among the supports of all the sources in the time-frequency domain. In fact, this assumption implies that sources are sparse in time-frequency domain. Under such assumption, each nonzero coefficient of STFT of the mixtures belongs to only one source. So we can unmix the sources by labeling the nonzero coefficients of STFT of the mixtures, then use inverse STFT to reconstruct the sources. Labeling is archived by using an attenuation *vs* delay histogram, in which each peak corresponds to one source.
- Brief introduction of Bofill's method
  As Bofill summarized in [18], the procedure is organized in three stages. First, the matrix of relative attenuations is inferred by angular clustering of the magnitude of the input, yielding a coarse partition of the data into their nearest sources. Second, for each partition, the differential delay is inferred by shifting the sensor channels and scattering the real and imaginary components until the cluster reappears. And third, given the attenuation and delay matrices, the sources are inferred at each frequency bin by assuming that the magnitude of the spectral coefficients is Laplacian distributed, which leads to the minimization of the sum of magnitudes, subject to the mixing equations. The resulting problem is an instance of second-order cone programming.

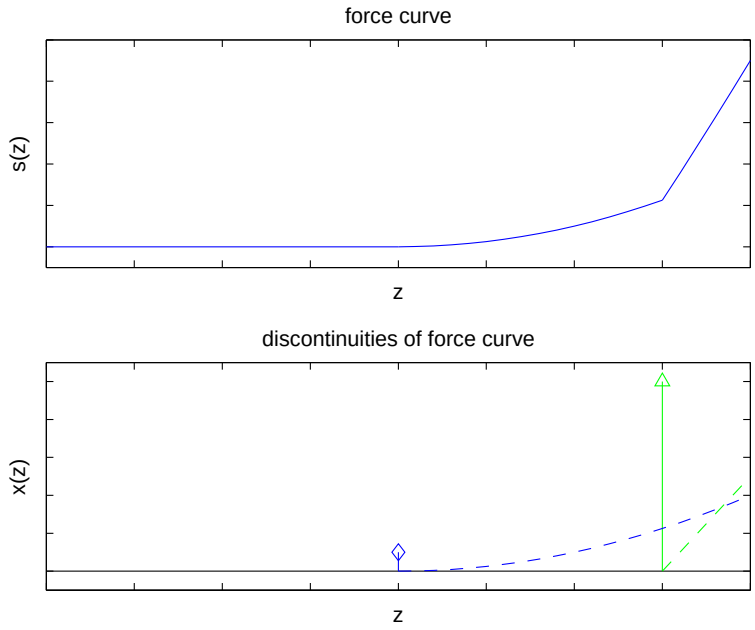


FIGURE 5.4 – Reconstruction of force curve. The upper subfigure is a force curve, and the lower subfigure are the discontinuities and their integrations. Diamond denotes the second order discontinuity and triangle denotes the first order discontinuity. The force curve can be reconstructed from its discontinuities by summing the integration of the first order discontinuity and the second order discontinuity.

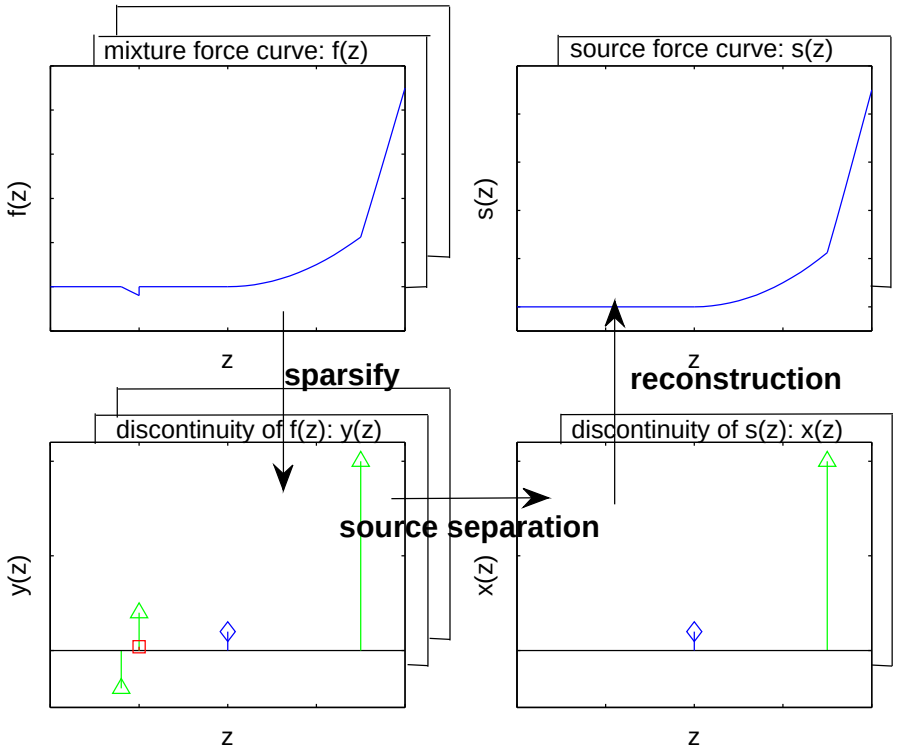


FIGURE 5.5 – Overall processing of AFM force curves. Square, triangle and diamond denote 0 1 and 2 order discontinuities respectively.

5.3 Identifiability

5.3.1 Modeling of sparse signals

In this subsection, we model the sources and mixtures in the transform domain. Though in the following texts, we deal with the force curves coming from AFM, which is piecewise polynomial signals, where the sources and mixtures are modeled by their discontinuities (locations, amplitudes and orders), it is worth pointing out that the modeling can be used to analysis signal more than piecewise polynomial signals. *e.g.,* piecewise polynomial basis can be replaced by other basis, discontinuity can be replaced by spike representing other quantity respectively, and the order of discontinuity can be replaced by other index which labels the membership of the spike in the subdictionaries, each subdictionary has the Toeplitz structure.

Firstly, we use $a_k\delta^{r_k}(z-z_k)$ to denote a r_k-th order discontinuity located at z_k with amplitude a_k. Then the sparse representation of the p-th source can be modeled as a discontinuity sequence, reads :

$$x_p(z) = \sum_{k=1}^{K_p} a_{k,p}\delta^{r_{k,p}}(z - z_{k,p}) \tag{5.9}$$

Remark 3. ***Initial conditions*** *: From Spline theory [182], the piecewise polynomial signal $f(z)$ is uniquely determined by its discontinuities $x(z)$ (knots in spline literature) and initial conditions, and can be reconstructed by integration (Fig. 5.4). From the knowledge of AFM, when the probe is far away from sample, the force is negligible, therefore it is reasonable to assume the initial conditions : $f(-\infty) = 0, f'(-\infty) = 0, f''(-\infty) = 0$.*

As is shown in [5], AFM force curve $f(z)$ can be modeled as the linear combination of shifted sources $s(z)$, *i.e.,* anechoic mixture, reads :

$$f_m(z) = \sum_{p=1}^{P_m} c_{p,m}s_p(z - \tau_{p,m}) \tag{5.10}$$

where $c_{p,m}$ is the weight of the p-th source and $\tau_{p,m}$ is the p-th shift. So we can also model the discontinuity sequence $y_m(z)$ (a sparse spike train) of force curve $f_m(z)$ (a continues signal) as the anechoic mixture of the discontinuity sequence $x_p(z)$ (a sparse spike train) of sources $s_p(z)$ (a continues signal) :

$$y_m(z) = \sum_{p=1}^{P_m} c_{p,m}x_p(z - \tau_{p,m}) \tag{5.11}$$

For notation simplicity, sometimes we omit the dependency on mixture index m in the following text unless when m is necessary. *e.g.,* $y(z)$ denotes the sparse representation of a mixture, while c_p denotes the contribution of p-th source in this mixture.

$y(z)$ can be estimated from $f(z)$ by using sparsify algorithms, *e.g.,* CSBR and the piecewise polynomial dictionary as was discussed in Chapter 2 and 3. The source separation problem in this chapter is then to estimate sources $x_p(z)$ and mixing parameters c_p, τ_p knowing $y(z)$. Afterwards, sources in original domain can be reconstructed from $x_p(z)$ following Remark 3. Fig. 5.5 shows the overall processing from mixture force curves to source force curves.

5.3.2 Ambiguities of source separation problem

In order to further discuss the identifiability, first let us start with the ambiguity. As a common problem in source separation [183], following ambiguities also exist in our problem :

- The energies of the sources cannot be determined. Because both mixing weights and sources are unknown, if we multiply one of them by a scalar α, then we can divide the other by the same scalar α, yielding the same mixtures. Sign ambiguity can be covered here as the special case where $\alpha = -1$;

- The shifts of the sources in each mixture cannot be determined. Because both shifts and sources are unknown, if we shift source by β, then we minus the shift by β, also yielding the same mixtures ;

- The order of the sources (source index p, distinguish with order of discontinuity r.) cannot be determined. Any permutation within indices of sources yields the same mixtures.

Remark 4. ***Scaling shifting and sorting :*** *Because of the ambiguities mentioned above, we cannot uniquely identify a source up to some scaling factor, shift and permutation. In order to have unicity, we always set the most left spikes of each source at location 0 with amplitude 1, and sort the sources by the location of their second spike. Sources with only one spike can be sorted according to their order information r.*

5.3.3 Definition of eigen interval

Before going further, let us define the eigen interval. By substituting (5.9) into (5.11) and omitting the dependency on m, we have :

$$y(z) = \sum_{p=1}^{P} c_p x_p(z - \tau_p) = \sum_{p=1}^{P} \sum_{k=1}^{K_p} c_p a_{k,p} \delta^{r_{k,p}}(z - z_{k,p} - \tau_p) \tag{5.12}$$

For a mixture $y(z) = \sum_{j=1}^{J} a_j \delta^{r_j}(z - z_j)$, if the j_1-th (j_2-th) spike $a_{j_1}\delta^{r_{j_1}}(z - z_{j_1})$ ($a_{j_2}\delta^{r_{j_2}}(z - z_{j_2})$) corresponds to the k_1-th (k_2-th) spike in the p_1-th (p_2-th) source, then we have :

$$\begin{aligned} a_{j_1} &= c_{p_1} a_{k_1,p_1} \\ r_{j_1} &= r_{k_1,p_1} \\ z_{j_1} &= z_{k_1,p_1} + \tau_{p_1} \end{aligned}$$

and

$$\begin{aligned} a_{j_2} &= c_{p_2} a_{k_2,p_2} \\ r_{j_2} &= r_{k_2,p_2} \\ z_{j_2} &= z_{k_2,p_2} + \tau_{p_2} \end{aligned}$$

If $p_1 = p_2 = p$, then we say the interval formed by spike j_1 and j_2 is an **eigen interval** : $\Delta z = z_{j_2} - z_{j_1} = z_{k_2,p} - z_{k_1,p}$, otherwise ($p_1 \neq p_2$) a **non-eigen interval** : $\Delta z = z_{j_2} - z_{j_1} = z_{k_2,p_2} - z_{k_1,p_1} + \tau_{p_2} - \tau_{p_1}$ because it depends on τ_{p_1} and τ_{p_2}. In a word, the two spikes forming an eigen interval coming from the the same source. If τ_{p_1} and τ_{p_2} are continuous random variable, the probability that this non-eigen interval coincides with other intervals is 0, this is our main assumption.

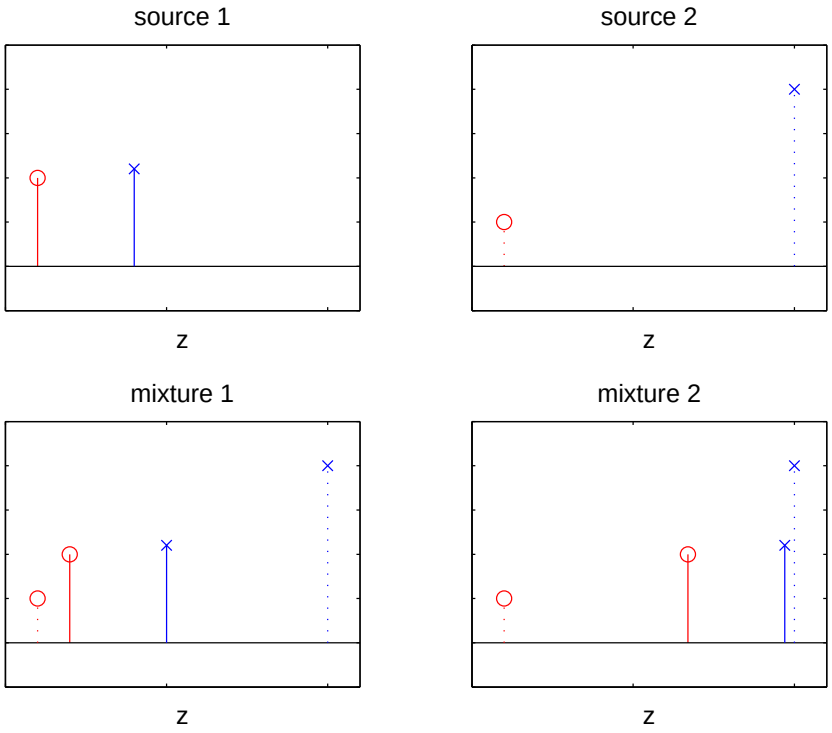


FIGURE 5.6 – An example shows the repeatability of eigen intervals. Circle (O) and cross (X) denote different orders of discontinuities. The eigen intervals : intervals in source 1 and source 2 present in both mixture 1 and mixture 2, while other non-eigen intervals do no. *e.g.,* the interval formed by the two circles in mixture 1 does not present in mixture 2.

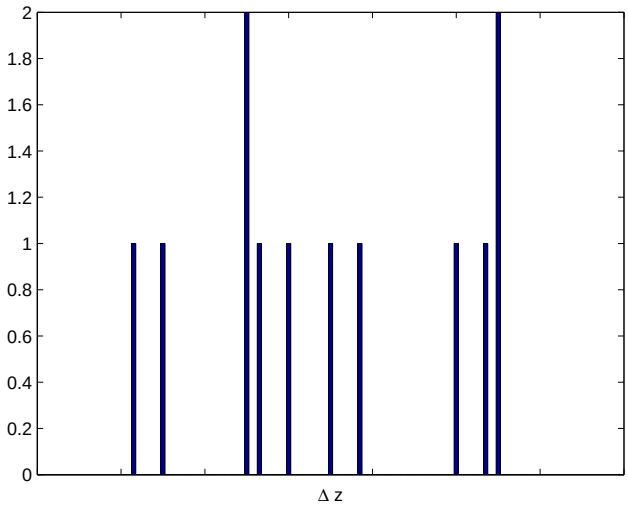


FIGURE 5.7 – Histogram of intervals of mixtures in Fig. 5.6. The two peaks correspond the eigen intervals in source 1 and source 2.

5.3.4 Sufficient condition for identifiability

Identifiability generally is referred to as the unique determination of the underlying parameters of the system from the observations. Precisely identifiability of the blind source separation problem means we can uniquely determine both the sources and mixing parameters given mixtures. In this part, we give a theoretical study in the **noise-free** setting.

Theorem 3. ***Sufficient condition :*** *If each source has at least two spikes, and presents at least twice in all the mixtures, each eigen interval is unique and each shift obeys continuous distribution, then the sparse source separation problem (5.11) has identifiability with probability equals to 1.*

Proof. Under the assumption that each source has at least two spikes, each source has at least one eigen interval. As we showed in the definition, the eigen interval is invariant w.r.t. the shifts. Under the assumption that each source presents at least twice in all the mixtures, in the histogram of all the intervals, *i.e.,* the intervals formed by all the combinations of two spikes in each mixture, the pillars of eigen intervals have height two at least (equal to the presenting times of the sources in all the mixtures). While the non-eigen interval is variant w.r.t. the shifts, under the assumption that each shift obeys continuous distribution, the probability that a non-eigen interval coincides with another non-eigen interval equals to 0, *i.e.,* each non-eigen interval is unique, so the pillar of non-eigen interval has height one. Thus the eigen intervals can be distinguished from non-eigen intervals by their pillars' heights. It is worth noticing that probability 0 does not mean impossible, but with Lebesgue measure 0. So the probability of exact detection of the eigen intervals equals to 1.

Once all eigen intervals are uniquely determined, then in the mixtures, we can use the eigen intervals to link the spikes coming from the same source, yielding the recover of sources. The 'link' means : if the interval formed by two spikes is an eigen interval, then we assign these two spikes to the same source. Under the assumption that each shift obeys continuous distribution, the probability that two spikes overlap the same location is 0, and under the assumption that each eigen interval is unique, each spike in each mixture can be assigned uniquely to one source, so for each mixture, all the linked spikes form a source estimate. And following Remark 4, the source estimates corresponding the same source can be merged, and the mixing parameters are uniquely determined, *e.g.,* if spike $a_j\delta^{r_j}(z - z_j)$ is the j-th spike in a mixture, and is identified as the most left spike in the p-th source, then the weight is a_j, and the shift is z_j.

In a word, both the sources and mixing parameters are uniquely determined, so the sparse separation problem has identifiability with probability equals to 1. □

Fig. 5.6 and Fig. 5.7 illustrate the invariance of the eigen intervals. The two eigen intervals corresponding the two sources present twice (once in mixture 1 and once in mixture 2), while other non-eigen intervals present only once. So we can easily distinguish the two eigen intervals from non-eigen intervals by the height of the pillars, and then use the detected two eigen intervals to link the solid spikes, yielding source 1, or link the dotted spikes, yielding source 2. Afterwards, the estimation of mixing parameters is obvious.

Remark 5. *When we use a finite number of piecewise polynomial signals to sparsify force curves, the estimated locations of discontinuities are discretized, yielding the discrete-valued intervals, and thus the probability of coincidence is not exact 0 but low, yielding that pillars of non-eigen intervals may have height more than 1. However large number of mixtures yields large number of presence of*

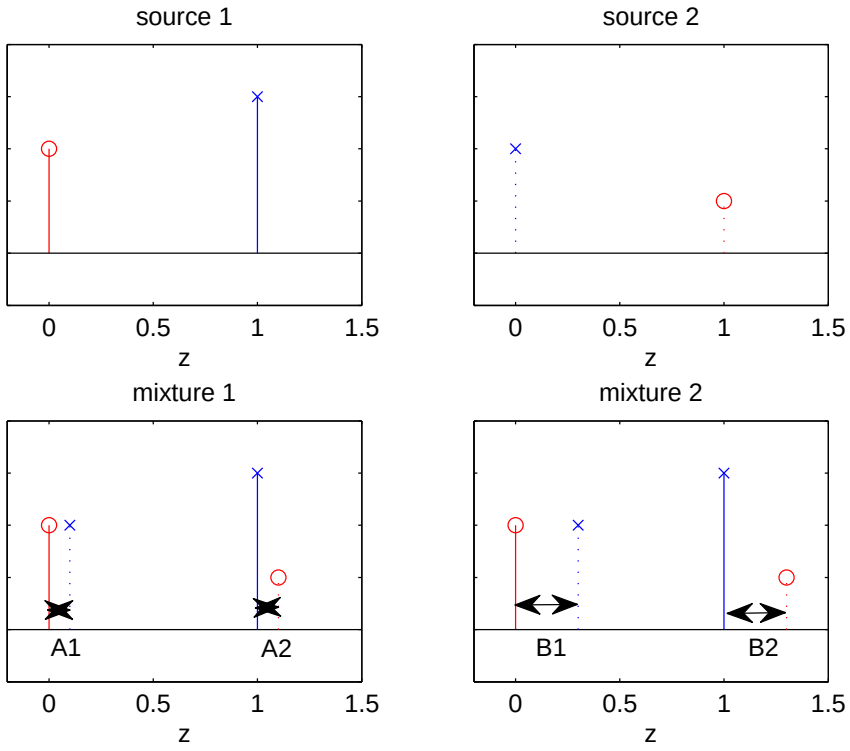


FIGURE 5.8 – An example where sources can be separated in 2D histogram but no in 1D histogram. Circle (O) and cross (X) denote 0 and 1 order discontinuities. Notice that both sources have interval equal to 1, so the two sources cannot be separated when order information is not taken into account, and *vice-versa*.

sources, thus guarantees the high enough pillars of eigen intervals which can be distinguished from the pillars of non-eigen intervals.

5.3.5 Extension of sufficient condition

Above discussion only considers the intervals of spikes. In fact when the order information of the spikes is taken into account, we can weaken the above discussed sufficient condition. The modification is : instead of 'each eigen interval is unique' in Theorem 3, we change it into 'each eigen interval is unique, or if the lengths of two eigen intervals are the same, but the differences between the order information of the left spike and that of the right spike are different.' *i.e.,* If one interval has order pair (r_l^1, r_r^1) and the other interval has order pair (r_l^2, r_r^2), then $r_l^1 - r_r^1 \neq r_l^2 - r_r^2$. Here **order pair** (r_l, r_r) indicates interval's left spike has order information r_l and right spike has order information r_r.

The proof is similar to that of Theorem 3. We use the example in Fig. 5.8, Fig. 5.9 and Fig. 5.10 to illustrate the extension. This time the two sources have the same interval length, so they cannot be separated in the 1D histogram, which is shown in Fig. 5.9. We can see the two real eigen intervals overlap in the histogram, and there are two false eigen intervals A and B. However, $r_l^1 - r_r^1 \neq r_l^2 - r_r^2$ can guarantee $(r_l^1, r_l^2) \neq (r_r^1, r_r^2)$, further $(r_l^{A1}, r_r^{A1}) = (r_l^1, r_l^2) \neq (r_r^1, r_r^2) = (r_l^{A2}, r_r^{A2})$, so interval $A1$ and $A2$ are separated into different histograms, *i.e.,* histograms titled (r_l^{A1}, r_r^{A1}) and (r_l^{A2}, r_r^{A2}), thus the two sources can be separated when order information is taken into account, yielding a 2D histogram in Fig. 5.10, each histogram titled (r_l, r_r) corresponds to the intervals with order pair (r_l, r_r). It is worth pointing out that Fig. 5.9 is in fact the sum of all histograms in Fig. 5.10. We call Fig. 5.9 a **1D histogram** while Fig. 5.10 a **2D histogram** since the latter has one more

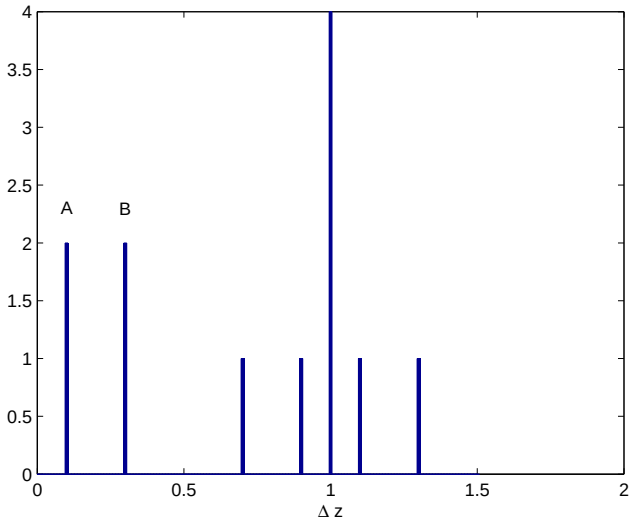


FIGURE 5.9 – Histogram of intervals of mixtures in Fig. 5.8. Order information is not taken into account. Source 1 and source 2 cannot be identified from this histogram because their intervals correspond the same bin located at $\Delta z = 1$.

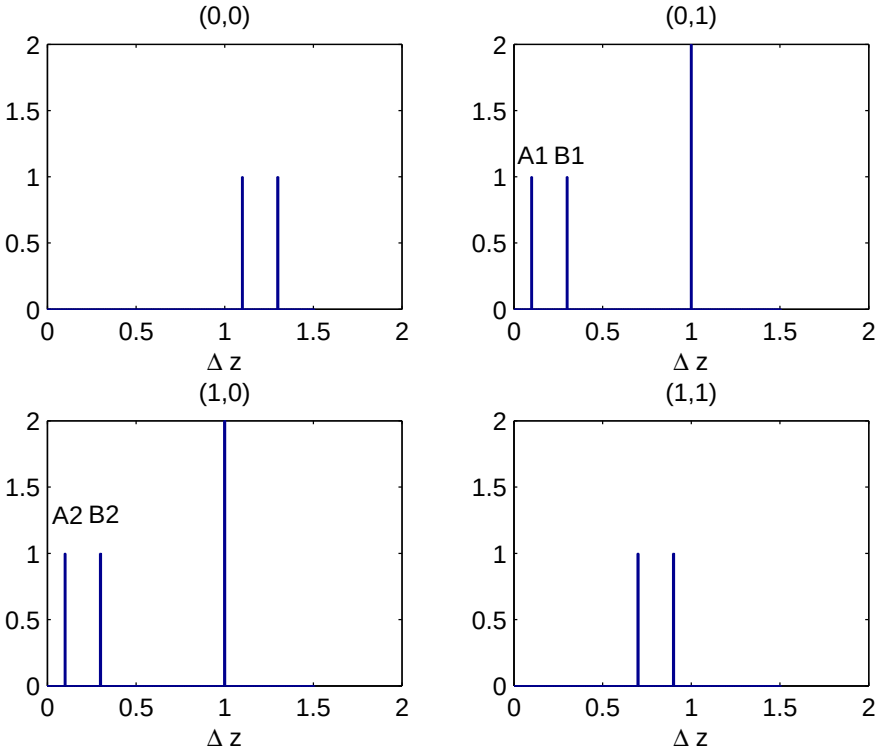


FIGURE 5.10 – Histograms of intervals of mixtures in Fig. 5.8. Order information is taken into account. The subfigure titled (r_l, r_r) presents the histogram of all the intervals in the mixtures, whose left spike is a r_l-th order discontinuity and the right spike is a r_r-th order discontinuity. Each bin with amplitude 2 represents one source. So source 1 can be detected in histogram $(0, 1)$ and source 2 can be detected in histogram $(1, 0)$.

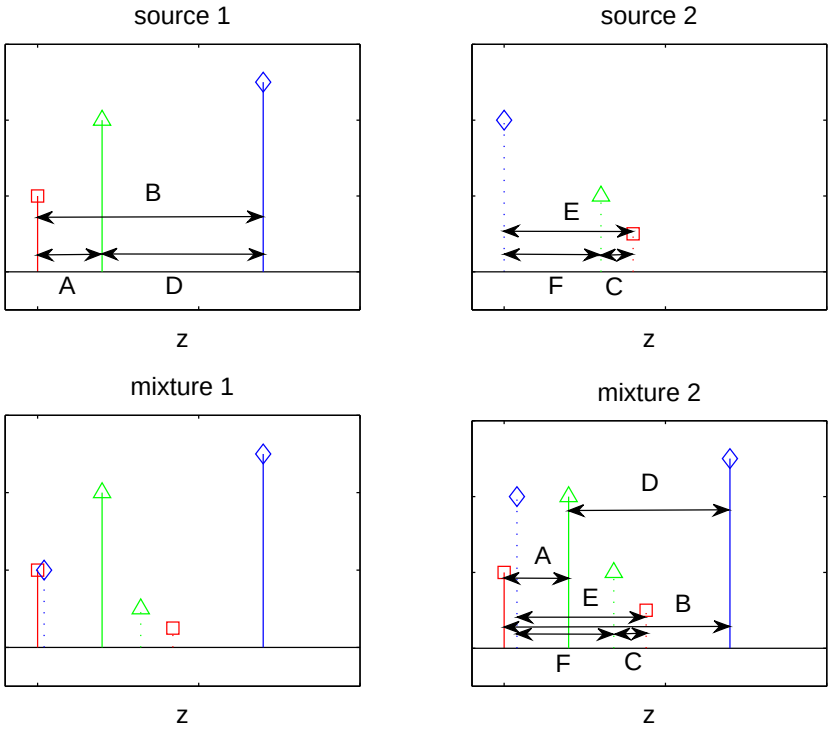


FIGURE 5.11 – An illustration where each source has three spikes. Square, triangle and diamond denote 0 1 and 2 order discontinuities respectively. The eigen intervals are labeled A∼F.

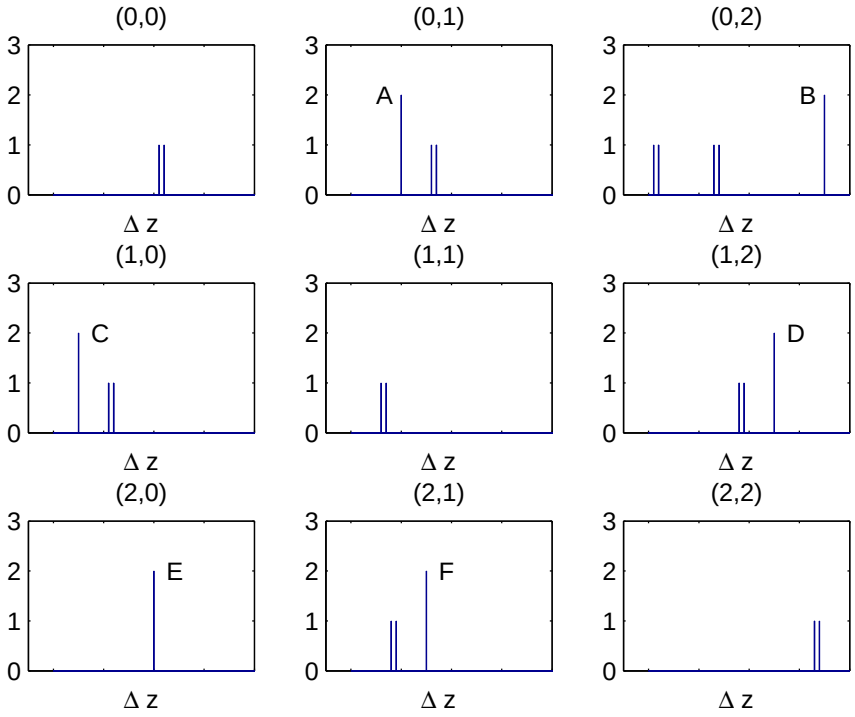


FIGURE 5.12 – Histograms of Fig. 5.11. Each bin with amplitude two represents one eigen interval labeled A∼F.

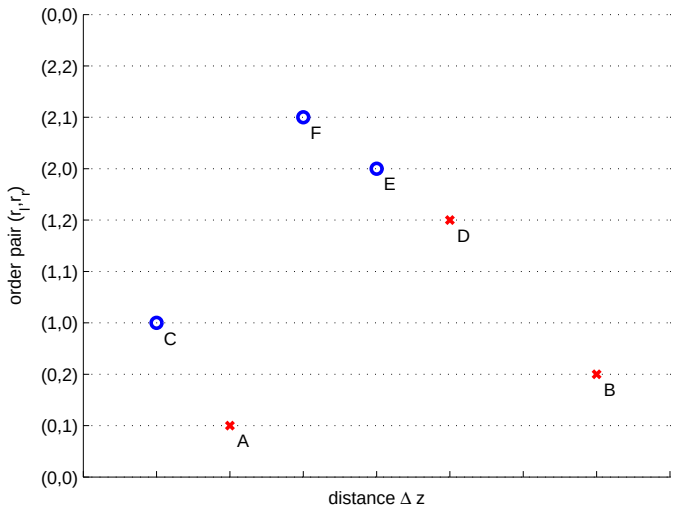


FIGURE 5.13 – Distance of discontinuity *vs* order of discontinuity domain distributions of two sources in Fig. 5.11. Crosses (X) represent source 1, circles (O) represent source 2. Each horizontal line of vertical coordinate (r_l, r_r) corresponds the histogram with title (r_l, r_r) in Fig. 5.12. Each mark labeled A∼F corresponds to one eigen interval.

dimension (r_l, r_r).

Fig. 5.11 shows an illustration, where each source has three discontinuities. Because the combination of the pair discontinuities is $\boldsymbol{C}_3^2 = 3$ in each source, there are totally $2 \times 3 = 6$ eigen intervals. Each discontinuity has order information $r \in \{0, 1, 2\}$, so $3^2 = 9$ possible configurations for (r_l, r_r), yielding nine histograms in Fig. 5.12 titled (r_l, r_r), each corresponds one configuration. We can see the six eigen intervals can be identified from the nine histograms as pillars with amplitude 2.

Remark 6. *Among all the eigen intervals coming from the same source, there are inherent relations. e.g., in example Fig. 5.11, A B and D belong to source 1, so length of B is the sum of that of A and D, A and B share the same r_l. We can check the relations in Fig. 5.13.*

Now we can explain in detail the **distance of discontinuity *vs* order of discontinuity** domain in Fig. 5.3. Axis *distance of discontinuity* denotes the interval of discontinuity : Δz. Axis *order of discontinuity* denotes all the configurations of order pair (r_l, r_r). Fig. 5.13 shows the distance of discontinuity *vs* order of discontinuity domain distributions of the two sources in Fig. 5.11. We can see this 2D image synthesize the nine 1D histograms, each horizontal line of vertical coordinate (r_l, r_r) corresponds the histogram with title (r_l, r_r), and each marker corresponds an eigen interval. We introduce this distance of discontinuity *vs* order of discontinuity domain in order to help the reader easier understand the distribution of eigen intervals and the identifiability. From Fig. 5.13, we can see the distribution of eigen intervals is disjoint, so this problem has identifiability.

5.4 Algorithm to separate sparse spike trains

In this section, we tackle the anechoic unmixing of sparse spike trains problem (5.11) : estimating sources $x_p(z)$ and mixing parameters $c_{p,m}, \tau_{p,m}$ knowing the mixtures $y_m(z)$. The key ideas of the algorithm is : for all sources, for all pair of spikes belonging to the same source, the number of occurrences of the pair in all the mixtures is expected to be large. Then, the idea is to make the list of all pairs of spikes in each mixture. A pair of spikes belonging to a given source will occur many times, while a pair of spikes belonging to different sources do not occur may times.

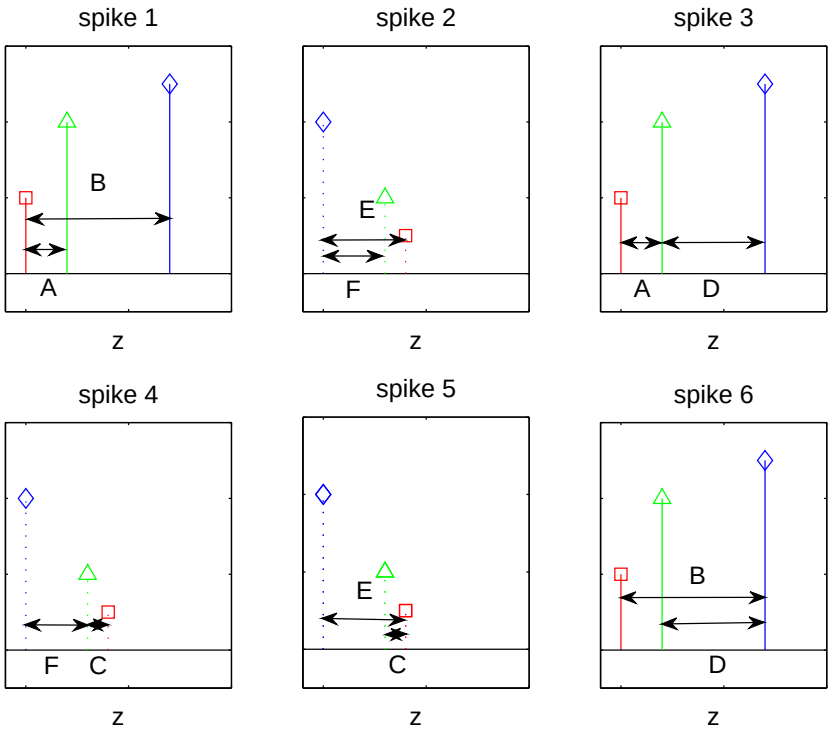


FIGURE 5.14 – Estimated sources from the label lists in Tab. 5.2. Spike 1, 3 and 6 belong to source 1, so they yield the same source (within some shifting and scaling) : source 1. While for spike 2, 4 and 5, the estimates are source 2. Each reference spike locates at 0 with amplitude 1.

5.4.1 Separate sparse spike trains in noise-free setting

The anechoic unmixing of sparse spike trains mainly consists five steps : (1) Estimating eigen intervals. (2) Labeling each spike in mixtures. (3) Fusing labels. (4) Estimating amplitude of spikes. (5) Estimating mixing parameters. In the first step, we detect and estimate the eigen intervals from the histograms, which is calculated from all the intervals of mixtures. In the second step, we use the estimated eigen intervals to match the intervals in the mixtures, and label the left and right spike of the matched intervals, yielding a label list for each spike in the mixtures. In the third step, we construct a source estimate from each label list by fusing the labels. Because the spikes from the same source yield the same source estimate, so after exploring all the label lists, we merge source estimates. In the fourth step, we propose a method to estimate the amplitude of spikes simultaneously with the estimation of locations. In the final step we use the successive cancelation algorithm to estimate the mixing parameters. In the following part, we introduce each procedure in detail.

Estimating eigen intervals

For each combination of spike pair from sparse spike train $y(z) = \sum_{j=1}^{J} a_j \delta^{r_j}(z - z_j), (z_1 < z_2 < \ldots < z_J, r_j \in \{0, 1, \ldots, R-1\})$, say the j_1-th spike $a_{j_1}\delta^{r_{j_1}}(z - z_{j_1})$ and j_2-th spike $a_{j_2}\delta^{r_{j_2}}(z - z_{j_2})$,$(j_1 < j_2)$, compute the 1D point $\Delta z = z_{j_2} - z_{j_1}$. $y(z)$ has J spikes, we can compute $\boldsymbol{C}_J^2 = \frac{J(J-1)}{2}$ points. And from M sparse spike trains, $\sum_{m=1}^{M} \boldsymbol{C}_{J_m}^2$ points totally. Then classify these points into groups w.r.t. order pair (r_{j_1}, r_{j_2}). $r_j \in \{0, 1, \ldots, R-1\}$, so order pair (r_{j_1}, r_{j_2}) has R^2 configurations. For each group, we draw 1D histogram of Δz with title (r_{j_1}, r_{j_2}), yielding R^2 1D histograms (see Fig. 5.12 where $R = 3$).

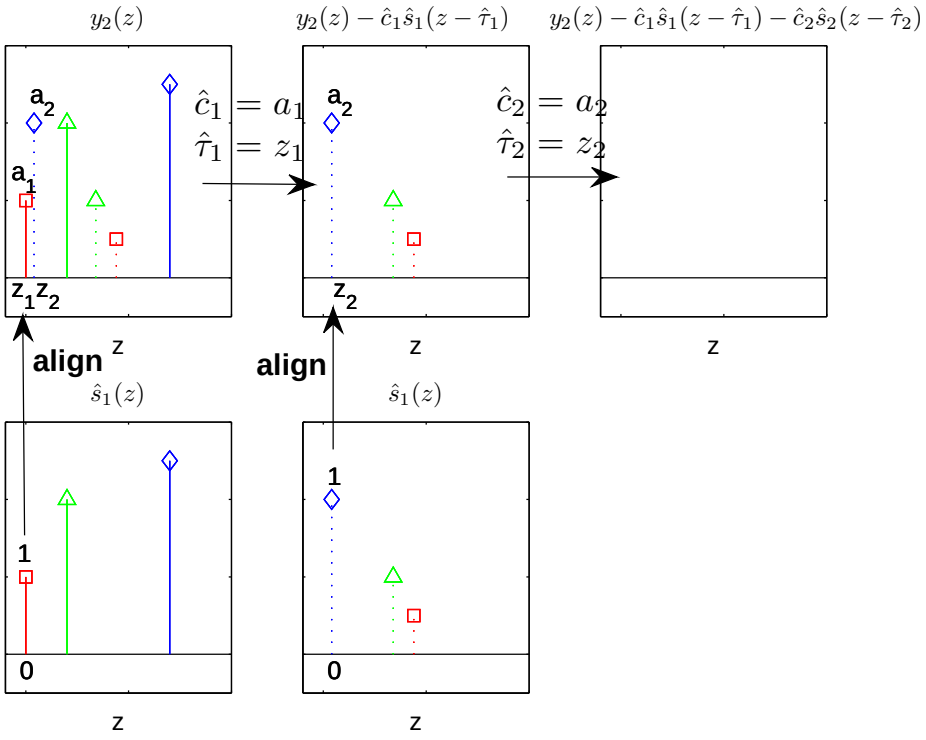


FIGURE 5.15 – Illustration of estimating mixing parameters by successive cancelation. $y_2(z)$ is the mixture 2 in Fig. 5.11. $\hat{s}_1(z)$ and $\hat{s}_2(z)$ are the two source estimates, whose most left spike located at 0 with amplitude 1. z_1, z_2, a_1 and a_2 are the locations and amplitudes of first and second spikes in mixture 2 respectively. $\hat{\tau}_1$, $\hat{\tau}_2$, $\hat{c}_1$ and $\hat{c}_2$ are the mixing parameter estimates.

As discussed in identification section, bins with height M correspond to eigen intervals, while other bins (most with height 1) correspond to non-eigen intervals, (see Fig. 5.12 where $M = 2$). As a result, eigen intervals can be detected as the bins with amplitude M, yielding N_{peak} estimates $\{\Delta\hat{z}_i | i = 1, \ldots, N_{peak}\}$ from R^2 histograms. Each estimated eigen interval $\Delta\hat{z}_i$ has one order pair $(\hat{r}_l^i, \hat{r}_r^i)$ inherited from the corresponding histogram title.

Remark 7. *When a source has J spikes, there are $\boldsymbol{C}_J^2 = \frac{J(J-1)}{2}$ pairs of spikes, yielding $\frac{J(J-1)}{2}$ eigen intervals, so in the histograms, $\frac{J(J-1)}{2}$ peaks correspond to this source.*

Labeling each spike in mixtures

Once we have eigen interval estimates $\{\Delta\hat{z}_i | i = 1, 2, \ldots, N_{peak}\}$, we can label each spike in mixtures by matching eigen intervals estimates with intervals in mixtures. *e.g.,* for spike pair j_1 and j_2 in mixture $y(z)$, if there exists i such that the interval of the spike pair ($\Delta z = z_{j_2} - z_{j_1}$) equals $\Delta\hat{z}_i$, and the order pair (r_{j_1}, r_{j_2}) equals that of $\Delta\hat{z}_i : (\hat{r}_l^i, \hat{r}_r^i)$, then we add label i^- to the label list of spike j_1, and label i^+ to the label list of spike j_2, which means spike j_1 sits on the left $(-)$ side and spike j_2 sits on the right $(+)$ side of the i-th estimated eigen interval $\Delta\hat{z}_i$. Thus we can assign each spike in each mixture with a label list $\{i^\bullet | i \in \{1, \ldots, N_{peak}\}, \bullet \in \{-, +\}\}$. Tab. 5.2 shows the label lists of spikes of mixture 2 in Fig. 5.11. We can see labels in the same list correspond the same source, so each label list can be used to recover one source.

TABLE 5.2 – Label lists of spikes of mixture 2 in Fig. 5.11. The eigen intervals are labeled A∼F.

spike(from left to right)	1	2	3	4	5	6
labels	A^-, B^-	E^-, F^-	A^+, D^-	F^+, C^-	E^+, C^+	B^+, D^+

Fusing labels

After labeling each spike in each mixture, we can construct a sparse spike train from each label list $\{i^\bullet | i \in \{1, \ldots, N_{peak}\}, \bullet \in \{-, +\}\}$. Firstly, we set one reference spike at location $z = 0$ with the order information equals that of the spike of this label list. Secondly for each negative (positive) label i^- (i^+) from this label list, we set one spike at location $z = \Delta\hat{z}_i$ ($z = -\Delta\hat{z}_i$) with order information $\hat{r}_l^i$ ($\hat{r}_r^i$). So from the label list containing $K - 1$ elements, we construct a sparse spike train $\hat{x}(z)$ with K spikes. Fig. 5.14 shows the estimated sources from the label lists in Tab. 5.2. We can see from spike 1, 3 and 6 we can recover source 1 and from spike 2, 4 and 5 we can recover source 2. This is because spikes from the same source yield the identical source estimate (within some shift and scaling). In order to merge the identical source estimates, Remark 4 should be recalled, where the most left spike always locates at 0. So finally we have to shift $\hat{x}(z)$ such that the most left spike locates at 0. Shifting is unnecessary when label list contains only negative labels i^-.

Remark 8. ***Single-spike source :*** *Until now, we can recover the sources with more than one spike. A spontaneous question is how to recover the single-spike source : source with only one spike ? In fact in a mixture, a non-labeled spike (label list is empty) comes from a single-spike source. So from each non-labeled spike, we can estimate one source with only one spike located at 0 with the same order information as the non-labeled spike. Because we have R different order information, so the maximal number of single-spike sources is R.*

Estimating amplitude of spikes

The amplitude of spikes can be estimated simultaneously during above procedures. When we compute a 1D point $\Delta z = z_{j_2} - z_{j_1}$, we can associate this 1D point with $\Delta a = \frac{a_{j_2}}{a_{j_1}}$ which is the relative amplitude. Then after we detect N_{peak} peaks in the histograms, yielding N_{peak} estimates $\{\Delta\hat{z}_i | i = 1, \ldots, N_{peak}\}$, we also associate each $\Delta\hat{z}_i$ with $\Delta\hat{a}_i$, which is the mean value of Δa whose associated 1D point Δz is in the bin of $\Delta\hat{z}_i$. This yields $\{\Delta\hat{a}_i | i = 1, \ldots, N_{peak}\}$. And finally, for a negative label i^- (positive label i^+), when we set a spike at location $\Delta\hat{z}_i$ ($-\Delta\hat{z}_i$) (before shifting the most left spike to location 0), we also set the amplitude equal to $\Delta\hat{a}_i$ ($\frac{1}{\Delta\hat{a}_i}$) simultaneously.

In order to merge the identical sources, the Remark 4 should be recalled again. Scaling is needed such that the most left spike of the estimated source has amplitude 1.

Estimating mixing parameters

Once the source estimates are known, we can estimate the mixing parameters : weights and delays. We use successive cancelation algorithm [184]. In each iterate, we align source estimates with the ongoing mixture by their most left spikes. Among all the source estimates, we choose the one that matches all the spikes in the ongoing mixture. *Match* means each spike in the source aligns a spike in the mixture and share the same order information. Once matched, we can estimate mixing parameters directly : the shift (weight) is the location (amplitude) of the most left spike in the ongoing mixture. Afterwards, cancel the matched spikes in the mixture, and then go to the most left spike of the remaining spikes for the next iterate. We repeat this procedure until all the spikes are canceled. Fig. 5.15 shows an illustration of estimating mixing parameters by successive cancelation.

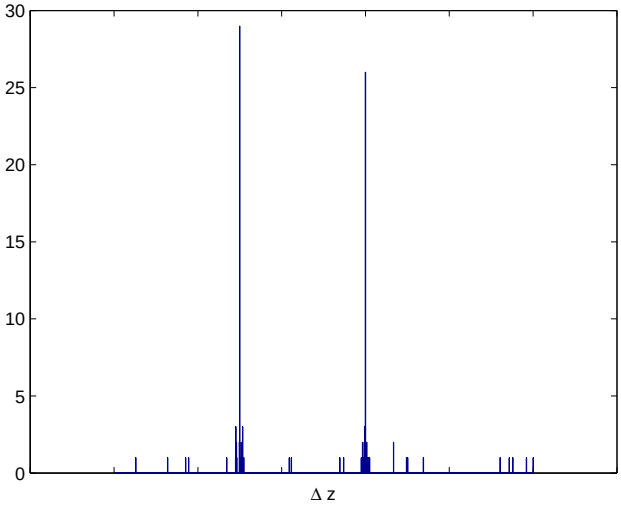


FIGURE 5.16 – Histogram of 30 noisy mixtures from two sources.

5.4.2 Extension to noisy setting

Differ from common definition, here *noisy* means there are perturbations in the locations of sparse spike train, which is due to the noise in data and sparsify algorithm.

The presence of noise yields that the bin height of eigen interval varies around M (Fig. 5.16). However, when the perturbation is moderate, the bins of eigen interval are still distinct from that of non-eigen intervals by their height. So we can detect the eigen intervals by thresholding the histograms with $M\lambda$, yielding estimates $\{\Delta\hat{z}_i\}$, each is the mean value of all the 1D points in the bin. Here we introduce the parameter λ to control the threshold.

Because of the presence of noise, we cannot match eigen interval in mixtures exactly, so modification in labeling procedure 5.4.1 is necessary : Instead of testing the exact equality between interval Δz of spike pair in mixture and $\Delta\hat{z}_i$, we now test the approximation within some tolerance : $|\Delta z - \Delta\hat{z}_i| \leq \epsilon$. Tolerance ϵ can be set, say 3σ (three-sigma rule), where σ is the standard deviation of the perturbation of the spike location. Same modification is also needed in mixing parameter estimation procedure 5.4.1, *match* should be defined as : each spike in the source align a spike in the mixture within some tolerance.

Because of the presence of noise, the spikes from the same source may not yield the identical source estimate, yielding more than P source estimates. So we can build a list of constructed sparse spike train, and choose the mostly frequent (large counter) P sparse spike train as the final source estimates. The list of sparse spike train is built as : Each time after construct a spike train from a label list, we query if the newly constructed sparse spike train (after shifting and rescaling) is already exist in the list. If the answer is true, then we just increase corresponding counter by one. Otherwise we append this newly constructed sparse spike train into the list, and initial the corresponding counter as 1.

5.5 Experiments

To evaluate the performance of our proposed algorithm, we run three simulations and one real data processing. In each simulation, we test several configurations of the parameters interesting according to the simulation objective. For each configuration, 1,000 or 10,000 Monte Carlo experiments are tested. For each experiment, we generate P sources following (5.9), each source has K

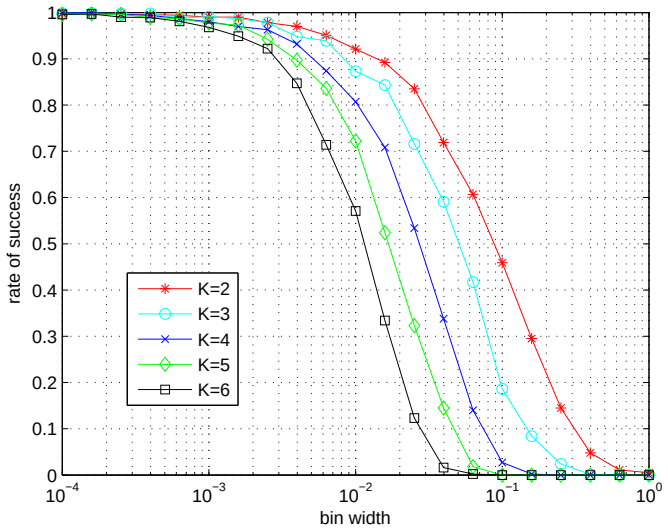


FIGURE 5.17 – Effect of bin width of histogram

spikes. The location $z_{k,p}$ of each spike has i.i.d. uniform distribution on continuous interval $[0, 1]$, the amplitude $a_{k,p}$ of each spike has i.i.d. Gaussian distribution $\mathcal{N}(\mu, \sigma_1^2)$ with mean $\mu = 0$ and variance $\sigma_1^2 = 0.01$, and the order information $r_{k,p}$ of each spike has i.i.d. uniform distribution on set $\{0, 1, 2\}$. Then the sources are shifted, scaled and sorted following Remark 4. M noise-free mixtures are generated following (5.11), mixing parameters $\tau_{p,m}$ and $c_{p,m}$ has i.i.d. Gaussian distribution $\mathcal{N}(0, 0.01)$.

5.5.1 Effect of bin width of histogram

In this simulation, we evaluate the effect of bin width ξ of histogram in the noise-free setting. This study is necessary because : the smaller the bin width is, the higher resolution can be achieved to distinguish two neighbour peaks in the histogram. Ideally, the smaller bin width is preferable. But too small bin width yields large number of bins, and thus needs too much (sometimes unnecessary) computation and memory. So there should be some compromise between resolution and computation, and we want to find small yet sufficient bin width.

The bin width increases from $1e-4$ to $1e-0$ with 20 uniform steps in the logarithmic scale. For each configuration of bin width, we run 1,000 Monte Carlo experiments. In each experiment, if all the sources estimates from two mixtures are equal to the original sources, we say this experiment is a *success*. Fig. 5.17 shows the rate of success w.r.t. the bin width.

We can see when the bin width is small, we have high probability to recover the sources. On the contrary, when the bin width is too large, we cannot recover the sources. This result agrees with our expectation. Among different configuration of sparsity level K (the number of spike in each source), if we want to keep the same rate of success, smaller bin width is required for larger K. This is because larger K means more eigen intervals, so smaller bin width is needed to increase the resolution of histogram. The conclusion of this simulation is : for K taking value from two to six, bin width $1e-4$ is small enough to recover the sources in noise-free setting.

5.5.2 Effect of perturbation on locations of spikes

In this simulation, we evaluate the performance of our proposed algorithm in the noisy setting, *i.e.,* we only focus on the perturbation on the location of spikes in mixtures. We assume that the

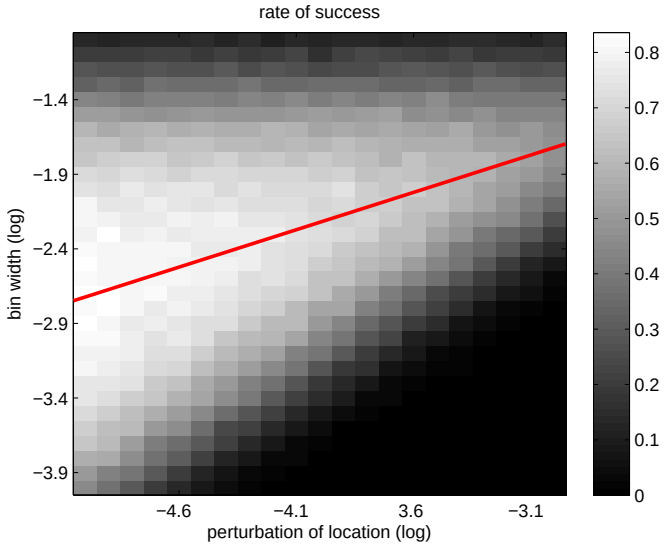


FIGURE 5.18 – Effect of spike location perturbation.

perturbation has i.i.d. Gaussian distribution $\mathcal{N}(0, \sigma_2^2)$, standard deviation σ_2 increases from $1e-5$ to $1e-3$ with 20 uniform steps in the logarithmic scale. For each perturbation level σ_2, the bin width increases from $1e-4$ to $1e-1$ with 30 uniform steps in the logarithmic scale. On the 2D *bin width vs perturbation* plane (Fig. 5.18), each pixel corresponds one configuration of parameter pair (σ_2, ξ). On each pixel, we run 1000 Monte Carlo experiments with sparsity level $K = 3$ and mixture number $M = 2$, and compute the rate of success. Because the presence of noise, the exact recovery of sources is almost impossible, as a result, the definition of *success* needed to be modified. We say one experiment is a success if the locations of each source estimate are close enough (inferior than $3\sigma_2$) to that of original source.

We can see the plane consists of three regions. In the upper region, we have low rate of success because the bin width is too large to distinguish nearby eigen intervals. In the middle region, we have high rate of success, especially on the line : $\log \xi = 0.5(\log \sigma_2 + 3) - 1.7$, *i.e.*, $\xi = 0.63\sqrt{\sigma_2}$, which crosses points $(\log \sigma_2 = -5, \log \xi = -2.7)$ and $(\log \sigma_2 = -3, \log \xi = -1.7)$. In the right lower region, we can hardly recover the sources, because the bin width is too small w.r.t. the perturbation level, the nonlinear denoising effect of the histogram is lost. *i.e.,* the perturbation of location of spikes yields the spreading of eigen intervals, too small bin width cannot cover all the spreaded eigen intervals, thus bin corresponding eigen interval does not have enough height needed to be detected. The conclusion of this simulation is : for given perturbation, there exists one bin width yielding best performance : $\xi = 0.63\sqrt{\sigma_2}$.

5.5.3 Effect of thresholding the histograms

When the noise presents in the mixtures, the bin corresponding to eigen intervals cannot reach M, we have to detect the eigen intervals by thresholding the histograms. In this simulation, we evaluate the effect of thresholding.

We run the simulation at three noise levels : $\sigma_2 = 1e-5, 1e-4, 1e-3$, and set corresponding tolerance $\epsilon = 3\sigma_2 = 3e-5, 3e-4, 3e-3$ respectively. For each σ_2, from Fig. 5.18, we find the best performance bin width is $\xi = 1e-2.7, 1e-2.2, 1e-1.7$ respectively. In order to observe the effect of thresholding w.r.t. mixture number M, we set $M = 20, 40, 60$. For each configuration of M, we run 10,000 Monte Carlo experiments. In each Monte Carlo experiment, we first generate $P = 2$ sources, each has $K = 3$ spikes, then generate M noise-free mixtures, finally generate noisy mixtures whose spike locations have perturbations of distribution $\mathcal{N}(0, \sigma_2^2)$. We increase threshold

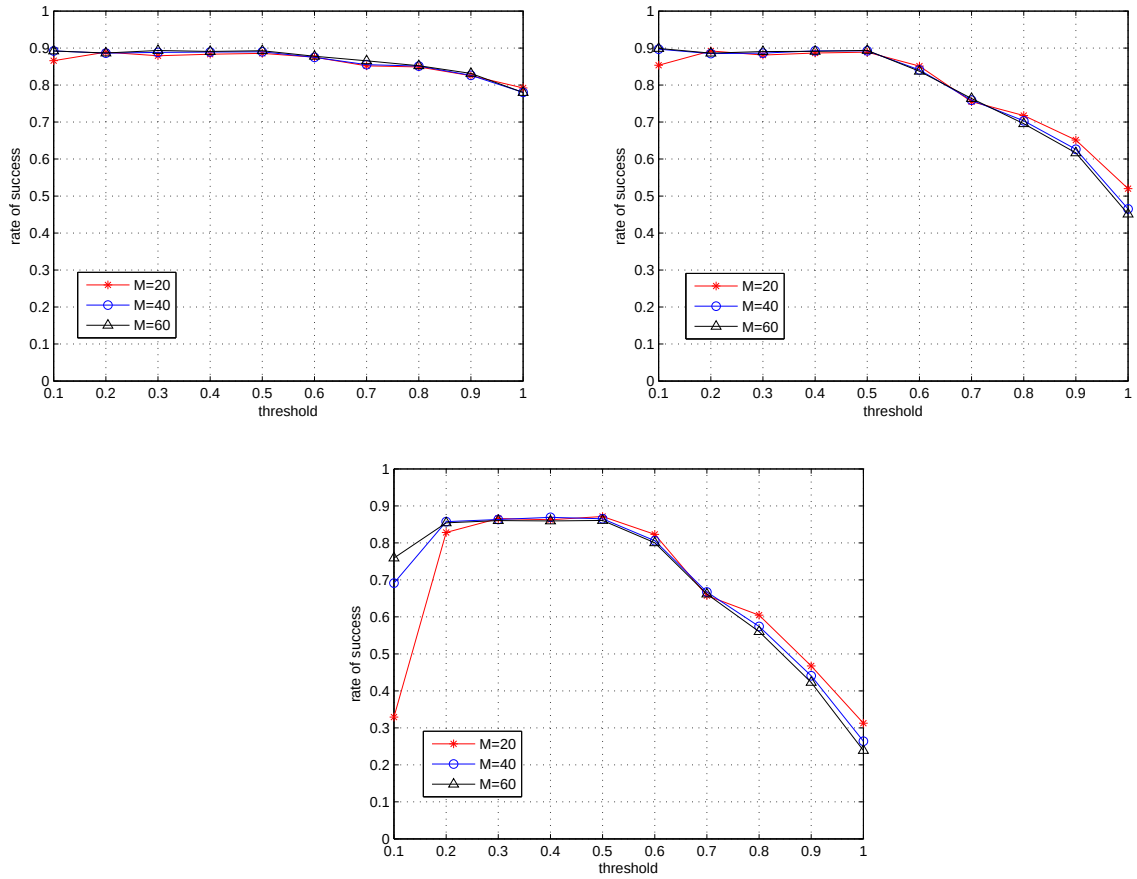


FIGURE 5.19 – Effect of thresholding. Upper left, upper right and lower figure has perturbation $\sigma_2 = 1e-5, 1e-4, 1e-3$ respectively.

parameter λ from 0.1 to 1 with step 0.1 for each Monte Carlo experiment. Fig. 5.19 shows the rate of success w.r.t. threshold parameter λ.

We can see at low noise level the performance varies very limited w.r.t. thresholding. While for higher noise level, threshold below $0.5M$ yields better performance. Threshold $0.5M$ is an interesting point, the performance start to degrade from this point. In fact, when a true eigen interval lies very close to a bin edge, because of the symmetric distribution of noise, two neighbour bins with height around $0.5M$ appear instead of one bin with height M. When threshold is below $0.5M$, this eigen interval cannot be detected. On the contrary, when the threshold increase above $0.5M$, the probability of missing detection increases. Nevertheless, we cannot decrease the threshold too low, otherwise false detection takes place. Results on configuration point $(M = 20, \lambda = 0.1)$ is such case, where $M\lambda = 2$. In a word, we propose the thresholding parameter $\lambda = 0.2 \sim 0.5$.

5.5.4 AFM data processing

In this experiment, we employ our proposed algorithm on AMF data measured from a staphylococcus lying on the glass slide. The 1024 approaching force curves are sampled on a 32×32 grid. In Fig. 5.20, we show 50 randomly selected force curves among 1024. We sparsity each force curve under the constraint that the approximation least square error is below $N\hat{\sigma}_n^2$, where $N = 500$ is the length of each force curve and $\hat{\sigma}_n^2$ is the noise variance estimated from the most left 100 points.

The histograms are shown in Fig. 5.21, where the bin width is 5nm. We can see the most significant peak sit in histogram (2,1) with height around 80. So we set threshing parameter $\lambda = 0.08$ in order

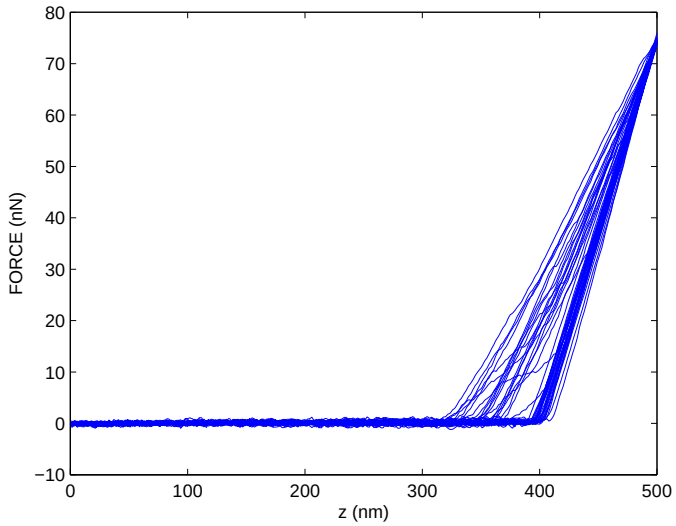


FIGURE 5.20 – 50 force curves measured from a staphylococcus lying on glass slide.

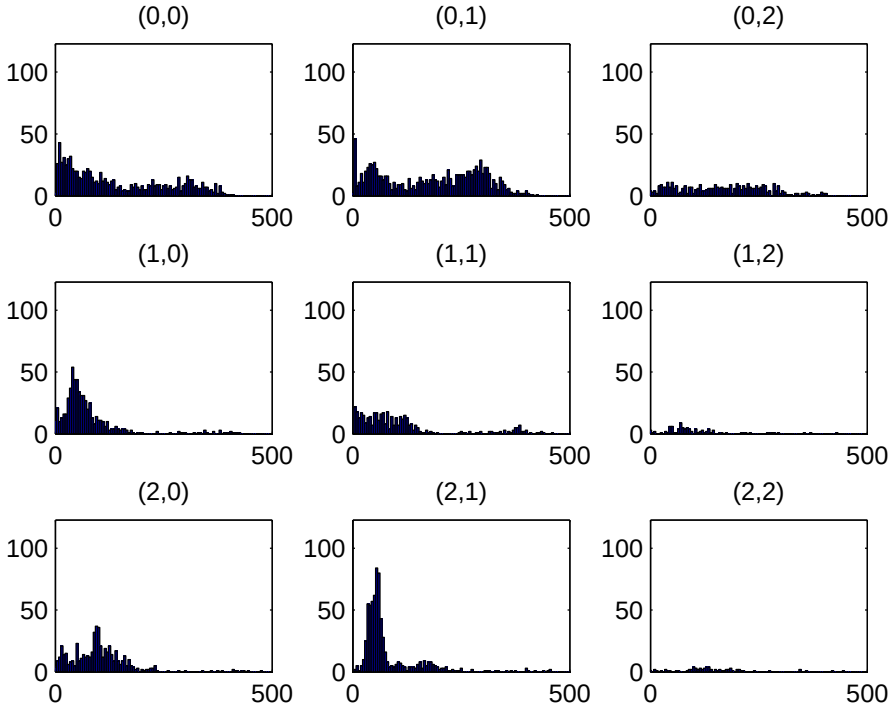


FIGURE 5.21 – Histograms of AFM data.

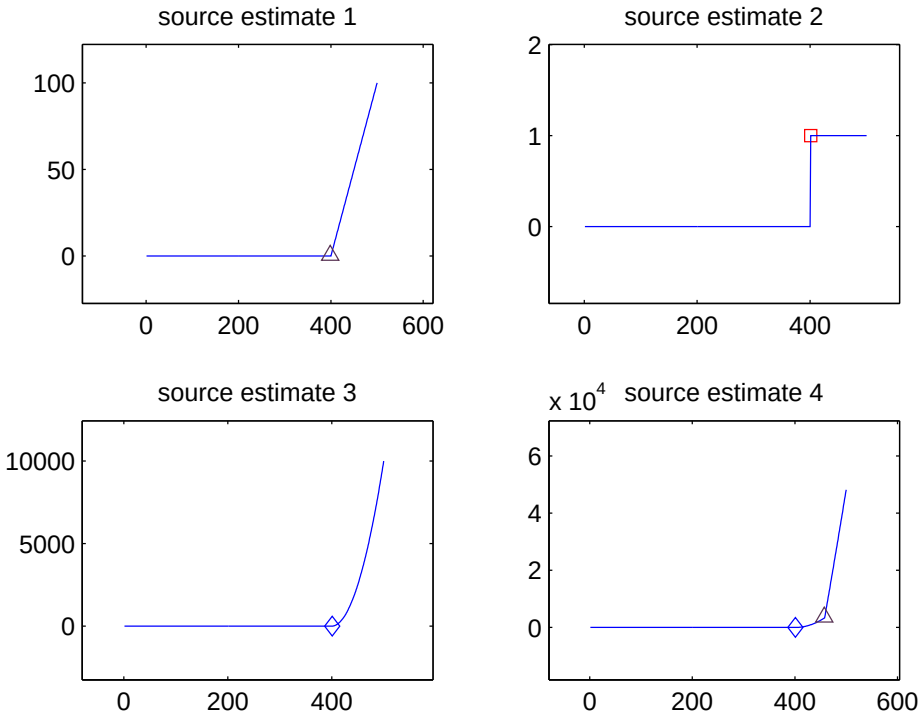


FIGURE 5.22 – Source estimates of AFM data. Square, triangle and diamond denote 0, 1 and 2 order discontinuities respectively.

to detect this peak only. The four source estimates and corresponding force curves are shown in Fig. 5.22. The first three source estimates have only one discontinuity (spikes), and the source estimate 4 has two discontinuities, the interval between the two discontinuities corresponds the peak in histogram (2,1). The second piece of the force estimate 4 is quadratic (the same as source estimate 3), which represents the elastic contact with staphylococcus, while the third piece is linear (the same as source estimate 1), which represents the rigid contact with glass slide. So force estimate 4 indicates the thin part of staphylococcus. Fig. 5.23 shows the mixing parameter estimates. Force curves on pixels in glass slide mainly consist of source 1 with shift 0 (see weight 1 and shift 1), while force curve on pixels in staphylococcus can be classified into two groups, one group corresponds source 3 (see weight 3), the other group corresponds source 4 (see weight 4). It is worth noticing that the large difference of amplitude of source estimates (source estimate 2 and 4) yields the large difference of weights (weight 2 and 4).

We run the processing on a Dell Precision T7400 workstation (4 core 3G Hz CPU, 3.4 GB memory). The sparsify algorithm (CSBR in Chapter 4) cost 200 seconds, sparse source separation algorithm cost 2 seconds.

5.6 Conclusion

In this chapter, we consider the joint processing of AFM force volume image. Starting from formulating each force curve as a mixture of shifted sparse sources, we proposed a method to separate a limited number of sources from a large number of shifted mixtures. The basic assumption is that each source can be sparsified into a sparse spike train for given dictionary, and that each eigen interval in source sparse spike train is unique. Based on these assumptions, we use histograms to accumulate the repeatability of eigen intervals. Once the eigen intervals are detected from the

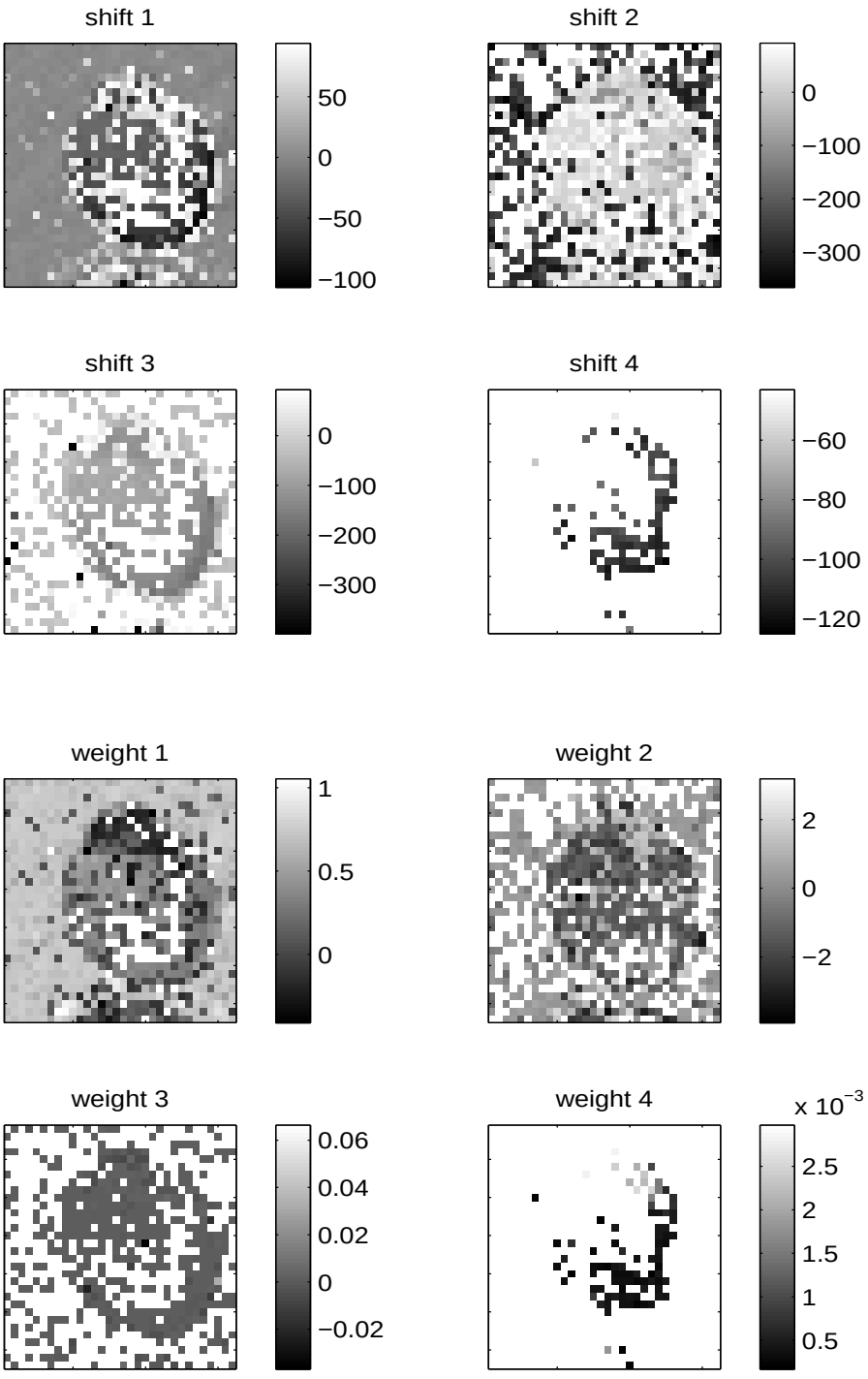


FIGURE 5.23 – Mixing parameter estimates of AFM data. Missing pixels are set infinite.

histograms, we can use them to link the spikes in the mixtures that belong to the same sources, yielding the source estimates. Once the source estimates are known, the estimation of mixing parameters is trivial. Experiments show the efficiency of our proposed method in both simulated and real data.

One limitation of the proposed algorithm is : because the number of columns in the piecewise polynomial signal dictionary is finite, the location of discontinuity is discrete-valued, yielding the estimation of shift is discrete-valued.

The performance of proposed method relies on the quality of histograms. We proposed the thresholding method to analysis the histograms. In the AFM data processing, peaks in histogram (1,0) and (2,0) (see Fig. 5.21) are also significant. However, we cannot detect them by decreasing the global thresholding parameter without introducing unnecessary false detection like the neighbour bins of the peak in histogram (2,1). In order to improve the performance, we should employ adaptive scattering method to segment each histogram, and then estimate the peaks in each segmentation. This is the first perspective for further study.

The second perspective is, our pre-defined piecewise polynomial dictionary is designed to sparsify the AFM force curves specially. However, when we want to apply our sparse source separation method to other real applications, *e.g.,* audio signal processing, the algorithm will fail because the piecewise polynomial dictionary does not yield sparse representation. As a result, it would be interesting to use machine learning to construct the data adaptive dictionary $\boldsymbol{A}$.

Conclusion

Les contributions de la thèse sont les suivantes. Du point de vue du traitement du signal, des algorithmes permettant l'approximation parcimonieuse d'un signal ont été développés, et appliqués au problème de la détection de discontinuités (détection d'un nombre limité de discontinuités dans un signal et approximation du signal par un signal polynômial par morceaux) et à la séparation de signaux polynômiaux par morceaux. Du point de vue appliqué, nous avons conçu des algorithmes de traitement automatique qui s'appuient sur la détection des discontinuités dans une courbe de force et sur les modèles physiques disponibles. Nous dressons un bilan succinct de cette thèse et nous envisageons quelques perspectives.

Approximation parcimonieuse

Dans un premier temps, nous avons établi un lien entre les algorithmes de restauration de processus Bernoulli-gaussien en traitement du signal et la minimisation de critères mixtes $\ell_2 - \ell_0$. L'algorithme SBR est défini comme une version limite de l'algorithme SMLR pour la restauration de processus Bernoulli-gaussiens, où la variance de la loi gaussienne tend vers $+\infty$. Il consiste à minimiser un critère $\ell_2 - \ell_0$ pénalisé $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{x}\|_0$ composé d'un terme d'attache aux données et d'un terme qui compte le nombre de variables sélectionnées. C'est un algorithme de type ajoute-retrait, proche d'autres algorithmes existants dans la littérature de la régression par sélection de variables. L'algorithme de continuation CSBR fournit un continuum de solutions heuristiques $x(\lambda)$ pour tout $\lambda \geq 0$. Les résultats obtenus sont très prometteurs sur des problèmes difficiles où les colonnes de la matrice d'observation $\boldsymbol{A}$ sont très corrélées.

Les algorithmes SBR et CSBR peuvent être interprétés comme des algorithmes sous-optimaux de minimisation discrète (le but étant de rechercher l'ensemble de colonnes à sélectionner dans la matrice d'observation $\boldsymbol{A}$ pour approcher au mieux un signal donné, le nombre de colonnes voulues étant connu ou non). Plusieurs perspectives sont envisagées :

1. L'étude théorique de la reconstruction exacte pour les algorithmes SBR et CSBR. Nous avons constaté empiriquement que sur certains types de problèmes simulés, en l'absence de bruit, le support de la solution $\boldsymbol{x}$ obtenue en sortie de SBR coïncide avec celui de la vraie solution $\boldsymbol{x}^*$ (à partir de laquelle les donnees $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^*$ ont été simulées) pourvu que λ soit suffisamment faible. Les simulations que nous avons effectuées sur un problème de déconvolution impulsionnelle (séparation de deux signaux gaussiens à supports non disjoints) montrent que sans bruit, SBR conduit à une reconstruction exacte dès lors que la valeur de λ est suffisamment faible, quel que soit la distance entre les deux signaux gaussiens.

   Ces résultats pratiques appellent une étude théorique (non encore réalisée), dans le but de confirmer que sous certaines hypothèses sur la matrice $\boldsymbol{A}$ et sur le signal $\boldsymbol{x}^*$ à reconstruire, SBR fournit la reconstruction exacte du signal inconnu $\boldsymbol{x}^*$ à partir de $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^*$. D'autres algorithmes possèdent cette propriété avec des conditions certes restrictives (les colonnes de $\boldsymbol{A}$ doivent être suffisamment décorrélées, et le vecteur $\boldsymbol{x}^*$ doit être suffisamment parcimonieux).
2. Étudier et comparer différentes stratégies automatiques pour la sélection de l'hyperparamètre

λ. La stratégie adoptée dans cette thèse repose sur la connaissance d'un seuil $\mathcal{E}_{min}$ sur l'erreur d'approximation $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$: lorsqu'une valeur de λ (ou de façon équivalente, lorsqu'un nombre k de colonnes sélectionnées par l'algorithme SBR) mène à une approximation dont l'erreur est inférieure à $\mathcal{E}_{min}$, CSBR s'arrête. Une solution alternative consiste à faire un choix automatique du nombre de colonnes k à sélectionner (l'ordre du modèle) sans nécessiter le réglage d'un seuil : le fait que CSBR estime un continuum de solutions $\boldsymbol{x}(\lambda)$ pour tout λ (ou de façon équivalente, $\boldsymbol{x}(k)$ pour tout k) rend possible la mise en œuvre, à moindre coût de calcul, de méthodes comme la courbe en L et l'optimisation de critères de sélection d'ordre comme le critère d'Akaike ou le critère *Minimum Description Length* (MDL). Ces stratégies mériteraient d'être testées et comparées.

Lissage d'un signal par un signal polynômial par morceaux

L'application des algorithmes de restauration de signaux parcimonieux au lissage d'un signal polynômial par morceaux peut être vue comme une stratégie de segmentation automatique des morceaux, à nombre de morceaux fixés ou à qualité d'approximation $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 \leq \mathcal{E}_{min}$ fixée.

En microscopie AFM, la recherche automatique des morceaux constitue une réelle difficulté lors du traitement d'une courbe de force. L'algorithme proposé est certainement une contribution non négligeable pour l'analyse pseudo-automatique d'une courbe de force. Rappelons que les analyses actuelles reposent essentiellement sur des règles de segmentation empiriques ou manuelles pour les régions d'intérêt.

Séparation de sources retardées

Nous avons proposé une stratégie en deux temps pour la séparation de sources retardées à partir d'un grand nombre de mélanges. Elle repose sur :

1. l'approximation parcimonieuse de chacun des mélanges $\boldsymbol{y}_i$ (avec un seuil commun sur l'erreur d'approximation $\|\boldsymbol{y}_i - \boldsymbol{A}\boldsymbol{x}_i\|^2 \leq \mathcal{E}_{min}$, ce qui implique que le nombre de colonnes de $\boldsymbol{A}$ nécessaires pour décrire chaque mélange dépend du contenu du signal $\boldsymbol{y}_i$) ;
2. la mise en correspondance des descriptions parcimonieuses de chacun des mélanges par un algorithme de recherche combinatoire. Cette étape repose sur une énumération de tous les couples de points de discontinuité (position et ordre de deux points de discontinuité du mélange) pour chaque mélange, et la recherche des couples redondants en termes d'ordres de discontinuité et de longueur de l'intervalle δz séparant les deux points de discontinuité.

Le fait de prendre en compte l'ordre des discontinuités en plus de leur position permet de rendre l'identifiabilité possible dans de nombreux cas. Des problèmes surviennent néanmoins lorsque plusieurs mélanges possèdent une seule discontinuité du même ordre (la notion de paire de points de discontinuité dans le mélange n'a plus de sens). Dans ce cas, il faut recourir à des règles ad hoc pour pouvoir apparier ou non les deux mélanges à la même source (source contenant un point de discontinuité) ou chaque mélange à une source distincte (chaque source contient aussi un point de discontinuité). Par exemple, on peut émettre des hypothèses sur les coefficients de mélange a_{ik} et sur les retards z_{ik} : on peut raisonnablement supposer que ces paramètres ont des variations spatiales douces (variations par rapport à la position i du pixel) et que pour certaines sources k, a_{ik} est nul dans des régions entières.

Dans la littérature, les contributions qui traitent de la séparation de sources parcimonieuses retardées concernent essentiellement un faible nombre de mélanges (souvent, 2 ou 3 mélanges) et reposent sur des hypothèses très fortes sur la parcimonie des signaux et sur les retards. Pour le problème de l'imagerie force-volume, ces hypothèses peuvent être relaxées car le nombre de mélanges

est très grand (il y a autant de mélanges que de pixels dans l'image) : notre hypothèse de travail (les signaux sources ont des dérivées parcimonieuses) est beaucoup moins restrictive que l'hypothèse de l'orthogonalité des sources dans le plan temps-fréquence, hypothèse habituelle pour la séparation de sources retardées à partir d'un faible nombre de mélanges.

Nous pensons que l'algorithme développé pourra donner lieu à des applications dans d'autres contextes que l'imagerie force-volume, comme dans des problèmes d'imagerie hyperspectrale.

Décomposition en motifs paramétriques

Une autre stratégie (non développée au cours de cette thèse) pour résoudre le problème de séparation de sources retardées en imagerie force-volume consiste à prendre en compte les modèles physiques qui décrivent les forces d'interaction entre une pointe et un échantillon homogène, que ce soit pour la phase d'approche (forces de van der Waals, électrostatiques, élastiques, *etc.*.) ou de retrait (forces d'adhésion, de capillarité, de liaison, *etc.*). Ces modèles, détaillés dans le chapitre 4 dans le cas de systèmes biologiques, consistent à décrire un signal source par un ensemble de modèles paramétriques définis chacun sur une région d'intérêt. Ce modèle s'écrit formellement sous la forme :

$$s_k(z) = \sum_{l=1}^{L} b_{kl} g_l(z; \boldsymbol{\theta}_{kl}) \tag{C.1}$$

où chaque fonction g_l décrit l'une des formes paramétriques présentes dans le signal source (par exemple une gaussienne tronquée, une discontinuité) et les paramètres $\boldsymbol{\theta}_{kl}$ associés représentent les facteurs de forme (typiquement la position, l'amplitude de la gaussienne et un paramètre de forme comme l'écart-type).

En intégrant le modèle paramétrique (C.1) dans le modèle de mélange de sources retardées, on obtient :

$$f(x_i, y_i, z) = \sum_{k=1}^{p} \sum_{l=1}^{L} a_{ik} b_{kl} g_l(z - z_{ik}; \boldsymbol{\theta}_{kl}) \tag{C.2}$$

Ce sont ces modèles g_l, qualifiés de *motifs élémentaires*, qu'il s'agira d'extraire d'une image force-volume par une procédure de décomposition.

Bibliographie

[1] R. Wiesendanger, *Scanning Probe Microscopy and Spectroscopy : Methods and Applications*, Cambridge Univ. Press, Cambridge, MA, 1994.

[2] H.-J. Butt, B. Cappella, and M. Kappl, "Force measurements with the atomic force microscope : technique, interpretation and applications", *Surface Science Reports*, vol. 59, no. 1–6, pp. 1–152, Oct. 2005.

[3] F. Gaboriaud and Y. F. Dufrêne, "Atomic force microscopy of microbial cells : application to nanomechanical properties, surface forces and molecular recognition forces", *Colloids and Surfaces B : Biointerfaces*, vol. 54, pp. 10–19, 2007.

[4] F. Gaboriaud and J.-J. Ehrhardt, "Effects of different crystal faces on surface charge of colloidal goethite (a-FeOOH) particles : an experimental and modeling study", *Geochimica and Cosmochimica Acta*, vol. 67, no. 5, pp. 967–983, 2003.

[5] C. Soussen, D. Brie, F. Gaboriaud, and C. Kessler, "Modeling of force-volume images in atomic force microscopy", in *Proc. IEEE ISBI*, Paris, May 2008.

[6] A. Blake, "Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-11, no. 1, pp. 2–12, Jan. 1989.

[7] S. Alliney and S. A. Ruzinsky, "An algorithm for the minimization of mixed l_1 and l_2 norms with application to Bayesian estimation", *IEEE Trans. Signal Processing*, vol. 42, no. 3, pp. 618–627, Mar. 1994.

[8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression", *Annals Statist.*, vol. 32, no. 2, pp. 407–451, 2004.

[9] P. J. Huber, *Robust Statistics*, John Wiley, New York, NY, 1981.

[10] D. M. Malioutov, M. Cetin, and A. S. Willsky, "Homotopy continuation for sparse signal representation", in *Proc. IEEE ICASSP*, Philadephia, PA, Mar. 2005, vol. V, pp. 733–736.

[11] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties", *J. Acoust. Society America*, vol. 96, no. 456, pp. 1348–1360, Dec. 2001.

[12] B. K. Natarajan, "Sparse approximate solutions to linear systems", *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, Apr. 1995.

[13] F. Champagnat, Y. Goussard, and J. Idier, "Unsupervised deconvolution of sparse spike trains using stochastic approximation", *IEEE Trans. Signal Processing*, vol. 44, no. 12, pp. 2988–2998, Dec. 1996.

[14] F. Champagnat, Y. Goussard, and S. G. Idier, "Déconvolution impulsionnelle", in *Approche bayésienne pour les problèmes inverses*, J. Idier, Ed., Paris, nov. 2001, pp. 115–138, Traité IC2, Série traitement du signal et de l'image, Hermès.

[15] A. J. Miller, *Subset Selection in Regression*, Chapman and Hall, London, UK, 2 edition, Apr. 2002.

[16] A. Kriznik, M. Bouillot, J. Coulon, and F. Gaboriaud, "Morphological specificity of yeast and filamentous Candida albicans forms on surface properties", *C. R. Biologies*, vol. 328, pp. 928–935, Aug. 2005.

[17] B. Rivet, L. Girin, and C. Jutten, "Solving the indeterminations of blind source separation of convolutive speech mixtures", in *Proc. IEEE ICASSP*, Philadelphia, PA, Mar. 2005, vol. 5, pp. 533–536.

[18] P. Bofill, "Underdetermined blind separation of delayed sound sources in the frequency domain", *Neurocomputing*, vol. 55, no. 3, pp. 627–641, Oct. 2003.

[19] C. Soussen, D. Brie, F. Gaboriaud, and C. Kessler, "Atomic force microscopy imaging and related issues in signal processing : a preliminary work", in *Institut franco-allemand pour les applications de la recherche*, Grenoble, France, Nov. 2007, IAR Workshop, pp. 1–12.

[20] V. Mazet, D. Brie, and J. Idier, "Decomposition of a chemical spectrum using a marked point process and a constant dimension model", in *Bayesian Inference and Maximum Entropy Methods*, A. Mohammad-Djafari, Ed., Paris, July 2006, MaxEnt Workshops.

[21] J. Duan, C. Soussen, D. Brie, and J. Idier, "A continuation approach to estimate a solution path of mixed L2-L0 minimization problems", in *Signal Processing with Adaptive Sparse Structured Representations (SPARS workshop)*, Saint-Malo, Apr. 2009, pp. 1–6.

[22] Y. Goussard, G. Demoment, and J. Idier, "A new algorithm for iterative deconvolution of sparse spike trains", in *Proc. IEEE ICASSP*, Albuquerque, NM, Apr. 1990, pp. 1547–1550.

[23] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification", *International Journal of Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[24] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries", *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[25] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit : Recursive function approximation with applications to wavelet decomposition", *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 40–44, 1993.

[26] J. Duan, C. Soussen, D. Brie, and J. Idier, "Détection conjointe de discontinuités d'ordres différents dans un signal par minimisation de critère L2-L0", in *Actes 22^e coll. GRETSI*, Dijon, France, Sep. 2009.

[27] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado, "Subset selection in noise based on diversity measure minimization", *IEEE Trans. Signal Processing*, vol. 51, no. 3, pp. 760–770, 2003.

[28] G. H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm", *IEEE Trans. Signal Processing*, vol. 57, no. 1, pp. 289–301, Jan. 2009.

[29] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit", *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[30] D. L. Donoho and Y. Tsaig, "Fast solution of l_1-norm minimization problems when the solution may be sparse", *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 4789–4812, Nov. 2008.

[31] J. J. Fuchs, "Recovery of exact sparse representations in the presence of bounded noise", *IEEE Trans. Inf. Theory*, vol. 51, no. 10, pp. 3601–3608, 2005.

[32] D. Needell and J. A. Tropp, "CoSaMP : Iterative signal recovery from incomplete and inaccurate samples", *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, Apr. 2008.

[33] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations", *The Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, Dec. 2008.

[34] C. Couvreur and Y. Bresler, "On the optimallity of the backward greedy algorithm for the subset selection problem", *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 3, pp. 797–808, Feb. 2000.

[35] D. Haugland, *A Bidirectional Greedy Heuristic for the Subspace Selection Problem*, vol. 4638 of *Lecture Notes in Computer Science*, pp. 162–176, Springer Verlag, Berlin, engineering stochastic local search algorithms. designing, implementing and analyzing effective heuristics edition, 2007.

[36] T. Zhang, "Adaptive forward-backward greedy algorithm for sparse learning with linear models", in *NIPS*, 2008, pp. 1921–1928.

[37] T. Blumensath and M. E. Davies, "On the difference between orthogonal matching pursuit and orthogonal least squares", Tech. Rep., University of Edinburgh, Mar. 2007.

[38] J. J. Kormylo and J. M. Mendel, "Maximum-likelihood detection and estimation of Bernoulli-Gaussian processes", *IEEE Trans. Inf. Theory*, vol. 28, pp. 482–488, 1982.

[39] J. M. Mendel, *Optimal Seismic Deconvolution*, Academic Press, New York, NY, 1983.

[40] Q. Cheng, R. Chen, and T.-H. Li, "Simultaneous wavelet estimation and deconvolution of reflection seismic signals", *IEEE Trans. Geosci. Remote Sensing*, vol. 34, pp. 377–384, Mar. 1996.

[41] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS : A re-weighted norm minimization algorithm", *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[42] S. Chen and J. Wigger, "Fast orthogonal least squares algorithm for efficient subset model selection", *IEEE Trans. Signal Processing*, vol. 43, no. 7, pp. 1713–1715, July 1995.

[43] R. Horn and C. Johnson, *Matrix analysis*, Cambridge University Press, 1985.

[44] S. J. Reeves, "An efficient implementation of the backward greedy algorithm for sparse signal reconstruction", *IEEE Signal Processing Letters*, vol. 6, pp. 266–268, Oct. 1999.

[45] S. F. Cotter, J. Adler, B. D. Rao, and K. Kreutz-Delgado, "Forward sequential algorithms for best basis selection", *IEE Proc. Vision, Image and Signal Processing*, vol. 146, no. 5, pp. 235–244, Oct. 1999.

[46] D. Ge, J. Idier, and E. L. Carpentier, "Enhanced sampling schemes for MCMC based blind Bernoulli-Gaussian deconvolution", Tech. Rep., Institut de Recherche en Communication et Cybernétique de Nantes, 2009.

[47] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, "Methods for modifying matrix factorizations", *Mathematics of Computation*, vol. 28, no. 126, pp. 505–535, Apr. 1974.

[48] C. Y. Chi and J. M. Mendel, "Improved maximum-likelihood detection and estimation of Bernoulli-Gaussian processes", *IEEE Trans. Inf. Theory*, vol. 30, pp. 429–435, 1984.

[49] M. Allain and J. Idier, "Efficient binary reconstruction for non-destructive evaluation using gammagraphy", *Inverse Problems*, vol. 23, no. 4, pp. 1371–1393, Aug. 2007.

[50] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation", *IEEE Trans. Signal Processing*, vol. 50, no. 6, pp. 1417–1428, 2002.

[51] F. R. Gantmacher and M. G. Krein, *Oscillation matrices and kernels and small vibrations of mechanical systems*, AMS Chelsea Publishing, 2002.

[52] F. Gaboriaud, B. S. Parcha, M. L. Gee, J. A. Holden, and R. A. Strugnell, "Spatially resolved force spectroscopy of bacterial surfaces using force-volume imaging", *Colloids and Surfaces B : Biointerfaces*, vol. 62, no. 2, pp. 206–213, Apr. 2008.

[53] C. Soussen, J. Duan, D. Brie, P. Polyakov, G. Francius, and J. Duval, "Force curve segmentation by piecewise polynomial approximation : mathematical formulation and complete structure of the algorithm", Tech. Rep., Centre de Recherche en Automatique de Nancy, 2010.

[54] E. Van den Berg, M. P. Friedlander, G. Hennenfent, F. J. Herrmann, R. Saab, and O. Yilmaz, "Algorithm 890 : Sparco : A testing framework for sparse reconstruction", *ACM Transactions on Mathematical Software*, vol. 35, no. 4, pp. 1–16, 2009.

[55] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling", *IEEE Signal Processing Magazine In Signal Processing Magazine*, pp. 21–30, 2008.

[56] P. Hansen, "Analysis of discrete ill-posed problems by means of the L-curve", *SIAM Rev.*, vol. 34, pp. 561–580, 1992.

[57] H. Akaike, "A new look at the statistical model identification", *IEEE Trans. Automat. Contr.*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.

[58] W. Paranchych and L. Frost, "The physiology and biochemistry of *pili*", *Adv. Microb. Physiol.*, vol. 29, pp. 53–114, 1988.

[59] R. Emerson and T. Camesano, "Nanoscale investigation of pathogenic microbial adhesion to a biomaterial", *Appl. Environ. Microbiol.*, vol. 70, pp. 6012–6022, 2004.

[60] P. N. Danese, L. A. Pratt, S. L. Dove, and R. Kolter, "The outer membrane protein, antigen 43, mediates cell-to-cell interactions within *Escherichia coli* biofilms", *Mol. Microbiol.*, vol. 37, pp. 424–432, 2000.

[61] M. Klausen, A. Heydorn, P. Ragas, L. Lambertsen, A. Aaes-Jøgensen, S. Molin, and T. Tolker-Nielsen, "Biofilm formation by *Pseudomonas aeruginosa* wild type, flagella and type IV pili mutants", *Mol. Microbiol.*, vol. 48, pp. 1511–1524, 2003.

[62] S. Da Re, B. L. Quéré, J.-M. Ghigo, and C. Beloin, "Tight modulation of *Escherichia coli* bacterial biofilm formation through controlled expression of adhesion factors", *Appl. Environ. Microbiol.*, vol. 73, pp. 3391–3403, 2007.

[63] N. Ledeboer and B. Jones, "Exopolysaccharide sugars contribute to biofilm formation by *Salmonella enterica serovar typhimurium* on HEp-2 cells and chicken intestinal epithelium", *J. Bacteriol.*, vol. 187, pp. 3214–3226, 2005.

[64] K. Jonas, H. Tomenius, A. Kader, S. Normark, U. Römling, L. M. Belova, and Öjar Melefors, "Roles of *curli*, cellulose and BapA in *Salmonella* biofilm morphology studied by atomic force microscopy", *BMC Microbiol.*, vol. 7, pp. 9, 2007.

[65] C. Beloin, A. Roux, and J.-M. Ghigo, "*Escherichia coli* biofilms.", *Curr. Top. Microbiol.*, vol. 322, pp. 249, 2008.

[66] H. Sun, D. R. Zusman, and W. Shi, "Type IV pilus of *Myxococcus xanthus* is a motility apparatus controlled by the frz chemosensory system", *Curr. Biol.*, vol. 10, pp. 1143–1146, 2000.

[67] M. McBride, "Bacterial gliding motility : Multiple mechanisms for cell movement over surfaces", *Annu. Rev. Microbiol.*, vol. 55, pp. 49–75, 2001.

[68] R. Harshey, "Bacterial motility on a surface : Many ways to a common goal", *Annu. Rev. Microbiol.*, vol. 57, pp. 249–273, 2003.

[69] L. Craig, M. E. Pique, and J. A. Tainer, "Type IV *pilus* structure and bacterial pathogenicity", *Nat. Rev. Microbiol.*, vol. 2, pp. 363–378, 2004.

[70] G. C. Ulett, A. N. Mabbett, K. C. Fung, R. I. Webb, and M. A. Schembri, "The role of F9 fimbriae of uropathogenic *Escherichia coli* in biofilm formation", *Microbiology*, vol. 153, pp. 2321–2331, 2007.

[71] A. N. Mabbett, G. C. Ulett, R. E. Watts, J. J. Tree, M. Totsika, C. lynn Y. Ong, J. M. Wood, W. Monaghan, D. F. Looke, G. R. Nimmo, C. Svanborg, and M. A. Schembri, "Virulence properties of asymptomatic bacteriuria *Escherichia coli*", *Int. J. Med. Microbiol.*, vol. 299, pp. 53–63, 2009.

[72] A. Alessandrini and P. Facci, "AFM : a versatile tool in biophysics", *Meas. Sci. Technol.*, vol. 16, pp. R65–R92, 2005.

[73] P. Schar-Zammaretti, "Surface morphology, elasticity and interactions of lactic acid bacteria studied by atomic force microscopy", *Abstr. Pap. Am. Chem. Soc.*, vol. 224, pp. U397–U397, 2002.

[74] Y. F. Dufrêne, C. Boonaert, H. C. van der Mei, H. J. Busscher, and P. G. Rouxhet, "Probing molecular interactions and mechanical properties of microbial cell surfaces by atomic force microscopy", *Ultramicroscopy*, vol. 86, pp. 113–120, 2001.

[75] M. Radmacher, "Measuring the elastic properties of living cells by the atomic force microscope", *Method Cell Biol.*, vol. 68, pp. 67–90, 2002.

[76] A. Engel and D. Muller, "Observing single biomolecules at work with the atomic force microscope", *Nat. Struct. Biol.*, vol. 7, pp. 715–718, 2000.

[77] G. Francius, B. Tesson, E. Dague, V. Martin-Jézéquel, and Y. Dufrêne, "Nanostructure and nanomechanics of live *Phaeodactylum tricornutum* morphotypes", *Environ Microbiol.*, vol. 10, pp. 1344–1356, 2008.

[78] G. Francius, S. Lebeer, D. Alsteens, L. Wildling, H. J. Gruber, P. Hols, S. D. Keersmaecker, J. Vanderleyden, and Y. F. Dufrêne, "Detection, localization, and conformational analysis of single polysaccharide molecules on live bacteria", *ACS Nano*, vol. 2, pp. 1921–1929, 2008.

[79] G. Francius, O. Domenech, M. Mingeot-Leclercq, and Y. Dufrêne, "Direct observation of *Staphylococcus aureus* cell wall digestion by lysostaphin", *J. Bacteriol.*, vol. 190, pp. 7904–7909, 2008.

[80] S. Velegol and B. Logan, "Contributions of bacterial surface polymers, electrostatics, and cell elasticity to the shape of AFM force curves", *Langmuir*, vol. 18, pp. 5256–5262, 2002.

[81] H. Clausen-Schaumann, M. Seitz, R. Krautbauer, and H. Gaub, "Force spectroscopy with single bio-molecules", *Curr. Opin. Chem. Biol.*, vol. 4, pp. 524–530, 2000.

[82] T. E. Fisher, P. E. Marszalek, and J. M. Fernandez, "Stretching single molecules into novel conformations using the atomic force microscope", *Nat. Struct. Biol.*, vol. 7, pp. 719–724, 2000.

[83] H. Janovjak, A. Kedrov, D. A. Cisneros, K. T. Sapra, J. Struckmeier, and D. J. Muller, "Imaging and detecting molecular interactions of single transmembrane proteins", *Neurobiol. Aging*, vol. 27, pp. 546–561, 2006.

[84] P. Hinterdorfer and Y. Dufrêne, "Detection and localization of single molecular recognition events using atomic force microscopy", *Nat. Methods*, vol. 3, pp. 347–355, 2006.

[85] Y. Dufrêne, "Towards nanomicrobiology using atomic force microscopy", *Nat. Rev. Microbiol.*, vol. 6, pp. 674–680, 2008.

[86] N. J. Tao, S. M. Lindsay, and S. Lees, "Measuring the microelastic properties of biological material", *Biophys. J.*, vol. 63, pp. 1165–1169, 1992.

[87] A. Vinckier and G. Semenza, "Measuring elasticity of biological materials by atomic force microscopy", *FEBS Lett.*, vol. 430, pp. 12–16, 1998.

[88] D. Ebenstein and L. Pruitt, "Nanoindentation of biological materials", *Nano Today*, vol. 1, pp. 26–33, 2006.

[89] J. Hoh and C. Schoenenberger, "Surface-Morphology and Mechanical-Properties of Mdck Monolayers by Atomic-Force Microscopy", *J. Cell Sci.*, vol. 107, pp. 1105–1114, 1994.

[90] M. Radmacher, M. Fritz, C. M. Kacher, J. P. Cleveland, and P. K. Hansma, "Measuring the viscoelastic properties of human platelets with the atomic force microscope", *Biophys. J.*, vol. 70, pp. 556–567, 1996.

[91] A. Engler, F. Rehfeldt, S. Sen, and D. Discher, "Microtissue elasticity : Measurements by atomic force microscopy and its influence on cell differentiation", *Method Cell Biol.*, vol. 83, pp. 521–545, 2007.

[92] S. Park, D. Koch, R. Cardenas, J. Käs, and C. K. Shih, "Cell motility and local viscoelasticity of fibroblasts", *Biophys. J.*, vol. 89, pp. 4330–4342, 2005.

[93] J. Domke, S. Dannöhl, W. J. Parak, O. Müller, W. K. Aicher, and M. Radmacher, "Substrate dependent differences in morphology and elasticity of living osteoblasts investigated by atomic force microscopy", *Colloids Surf.*, vol. 19, pp. 367–379, 2000.

[94] V. Vadillo-Rodriguez, T. J. Beveridge, and J. R. Dutcher, "Surface viscoelasticity of individual gram-negative bacterial cells measured using atomic force microscopy", *J. Bacteriol.*, vol. 190, pp. 4225–4232, 2008.

[95] A. Touhami, B. Nysten, and Y. F. Dufrêne, "Nanoscale mapping of the elasticity of microbial cells by atomic force microscopy", *Langmuir*, vol. 19, pp. 4539–4543, 2003.

[96] M. Beckmann, S. Venkataraman, M. Doktycz, J. Nataro, C. Sullivan, J. Morrell-Falvey, and D. Allison, "Measuring cell surface elasticity on enteroaggregative *Escherichia coli* wild type and dispersin mutant by AFM", *Ultramicroscopy*, vol. 106, pp. 695–702, 2006.

[97] E. K. Dimitriadis, F. Horkay, J. Maresca, B. Kachar, and R. S. Chadwick, "Determination of Elastic Moduli of Thin Layers of Soft Material Using the Atomic Force Microscope", *Biophys. J.*, vol. 82, pp. 2798–2810, 2002.

[98] G. Francius, J. Hemmerlé, J. Ohayon, P. Schaaf, J. Voegel, C. Picart, and B. Senger, "Effect of crosslinking on the elasticity of polyelectrolyte multilayer films measured by colloidal probe AFM", *Microsc Res Tech*, vol. 69, pp. 84–92, 2006.

[99] J. F. Duval, G. Francius, P. Polyakov, J. Merlin, Y. Abe, J.-M. Ghigo, C. Merlin, and C. Beloin, "Bacterial surface appendages strongly impact nanomechanical and electrokinetic properties of *Escherichia coli* cells subjected to osmotic stress.", *Prog. Biophys. Mol. Biol.*, 2010.

[100] M. Mustata, K. Ritchie, and H. A. McNally, "Neuronal elasticity as measured by atomic force microscopy", *J. Neurosci. Methods*, vol. 186, pp. 35–41, 2010.

[101] X. Chen, L. Feng, H. Jin, S. Feng, and Y. Yu, "Quantification of the erythrocyte deformability using atomic force microscopy : Correlation study of the erythrocyte deformability with atomic force microscopy and hemorheology", *Clin. Hemorheol. Microcirc.*, vol. 43, pp. 241–249, 2009.

[102] S. E. Cross, Y.-S. Jin, J. Rao, and J. K. Gimzewski, "Nanomechanical analysis of cells from cancer patients", *Nat. Nanotechnol.*, vol. 2, pp. 780–783, 2007.

[103] G. Lee, E. Park, S. Lee, J. Lim, Y. Eo, J. Han, S. Choi, J. Park, Y. Shin, H. Jin, K. Hong, B. Oh, and H. Park, *Measurement of Structural and Mechanical Property of Live Mesangial Cell (MC) by Atomic Force Microscopy (AFM)*, Crc Press-Taylor and Francis Group, 2009.

[104] S. E. Cross, Y.-S. Jin, J. Tondre, R. Wong, J. Rao, and J. K. Gimzewski, "AFM-based analysis of human metastatic cancer cells", *Nanotechnology*, vol. 19, 2008.

[105] B. Bhushan and N. Chen, "AFM studies of environmental effects on nanomechanical properties and cellular structure of human hair", *Ultramicroscopy*, vol. 106, pp. 755–764, 2006.

[106] R. La and M. Arnsdorf, "Multidimensional atomic force microscopy for drug discovery : A versatile tool for defining targets, designing therapeutics and monitoring their efficacy", *Life Sci.*, vol. 86, pp. 545–562, 2010.

[107] F. Malfatti and F. Azam, "Atomic force microscopy reveals microscale networks and possible symbioses among pelagic marine bacteria", *Aquat. Microb. Ecol.*, vol. 58, pp. 1–14, 2009.

[108] W. Tang and B. Bhushan, "Adhesion, friction and wear characterization of skin and skin cream using atomic force microscope", *Colloid Surf. B-Biointerfaces*, vol. 76, pp. 1–15, 2010.

[109] S. Ramachandran, A. P. Quist, S. Kumar, and R. Lal, "Cisplatin nanoliposomes for cancer therapy : AFM and fluorescence Imaging of cisplatin encapsulation, stability, cellular uptake, and toxicity", *Langmuir*, vol. 22, pp. 8156–8162, 2006.

[110] S. Sen and S. Kumar, "Combining mechanical and optical approaches to dissect cellular mechanobiology", *J. Biomech.*, vol. 43, pp. 45–54, 2010.

[111] M. Lekka and P. Laidler, "Applicability of AFM in cancer detection", *Nat. Nanotechnol.*, vol. 4, pp. 72–72, 2009.

[112] M. Lekka and J. Wiltowska-Zuber, "Biomedical applications of AFM", in *Nano 2008 : 2nd National Conference on Nanotechnology*, 2009, pp. 12023–12023.

[113] H. Jin, X. Xing, H. Zhao, Y. Chen, X. Huang, S. Ma, H. Ye, and J. Cai, "Detection of erythrocytes influenced by aging and type 2 diabetes using atomic force microscope", *Biochem. Biophys. Res. Commun.*, vol. 391, pp. 1698–1702, 2010.

[114] S. E. Cross, Y.-S. Jin, J. Rao, and J. K. Gimzewski, "Applicability of AFM in cancer detection Reply", *Nat. Nanotechnol.*, vol. 4, pp. 72–73, 2009.

[115] C. K. M. Fung, K. Seiffert-Sinha, K. W. C. Lai, R. Yang, D. Panyard, J. Zhang, N. Xi, and A. A. Sinha, "Investigation of human keratinocyte cell adhesion using atomic force microscopy", *Nanomed.-Nanotechnol. Biol. Med.*, vol. 6, pp. 191–200, 2010.

[116] S. Kasas and G. Dietler, "Probing nanomechanical properties from biomolecules to living cells", *Pflugers Arch.*, vol. 456, pp. 13–27, 2008.

[117] D. Lin, E. Dimitriadis, and F. Horkay, "Robust strategies for automated AFM force curve analysis -I : Non-adhesive indentation of soft, inhomogeneous materials", *J. Biomech. Eng.-Trans. ASME*, vol. 129, pp. 430–440, 2007.

[118] D. Lin, E. Dimitriadis, and F. Horkay, "Robust strategies for automated AFM force curve analysis -II : Adhesion-influenced indentation of soft, elastic materials", *J. Biomech. Eng.-Trans. ASME*, vol. 129, pp. 904–912, 2007.

[119] D. Rudoy, S. G. Yuen, R. D. Howe, and P. J. Wolfe, "Bayesian changepoint analysis for atomic force microscopy and soft material indentation", *J Roy Stat Soc C-App*, vol. 59, pp. 573–593, 2010.

[120] S. Sen, S. Subramanian, and D. E. Discher, "Indentation and adhesive probing of a cell membrane with AFM : Theoretical model and experiments", *Biophys. J.*, vol. 89, pp. 3203–3213, 2005.

[121] M. W. Rutland, J. G. Tyrrell, and P. Attard, "Analysis of atomic force microscopy data for deformable materials", *Journal of Adhesion Sciences Technology*, vol. 18, pp. 1199–1215, 2004.

[122] P. Attard and J. Parker, "Deformation and adhesion of elastic bodies in contact", *Phys. Rev. A*, vol. 46, pp. 7959–7971, 1992.

[123] J. Song, D. Tranchida, and G. J. Vancso, "Contact mechanics of UV/ozone-treated PDMS by AFM and JKR testing : Mechanical performance from nano- to micrometer length scales", *Macromolecules*, vol. 41, pp. 6757–6762, 2008.

[124] D. C. Lin, E. K. Dimitriadis, and F. Horkay, "Elasticity of rubber-like materials measured by AFM nanoindentation", *Express Polym. Lett.*, vol. 1, pp. 576–584, 2007.

[125] S. G. Yuen, D. Rudoy, R. D. Howe, and P. J. Wolfe, "Bayesian changepoint detection through switching regressions : Contact point determination in material indentation experiments", *Statistical Signal Processing, 2007. SSP '07. IEEE/SP 14th Workshop*, pp. 104–108, 2007.

[126] S. Cui, Y. Yu, and Z. Lin, "Modeling single chain elasticity of single-stranded DNA : A comparison of three models", *Polymer*, vol. 50, pp. 930–935, 2009.

[127] T. Camesano and N. Abu-Lail, "Heterogeneity in bacterial surface polysaccharides, probed on a single-molecule basis", *Biomacromolecules*, vol. 3, pp. 661–667, 2002.

[128] F. Gaboriaud, M. L. Gee, R. Strugnell, and J. F. L. Duval, "Coupled Electrostatic, Hydrodynamic, and Mechanical Properties of Bacterial Interfaces in Aqueous Media", *Langmuir*, vol. 24, pp. 10988–10995, 2008.

[129] H. Ohshima, "Electrostatic interaction between soft particles", *J. Colloid Interface Sci.*, vol. 328, pp. 3–9, 2008.

[130] J. Lyklema, *Fundamentals of Interface and Colloid Science*, Elsevier/Academic Press, 2005.

[131] M. Giesbers, *Surface forces studied with colloidal probe atomic force microscopy*, Wageningen University, 2001.

[132] H. Hertz, "Ueber die Berührung fester elastischer Körper", *J Reine Angew Math*, vol. 92, pp. 156–171, 1881.

[133] X. Yao, J. Walter, S. Burke, S. Stewart, M. H. Jericho, D. Pink, R. Hunter, and T. J. Beveridge, "Atomic force microscopy and theoretical considerations of surface properties and turgor pressures of bacteria", *Colloid Surf. B-Biointerfaces*, vol. 23, pp. 213–230, 2002.

[134] C. Ortiz and G. Hadziioannou, "Entropic elasticity of single polymer chains of poly(methacrylic acid) measured by atomic force microscopy", *Macromolecules*, vol. 32, pp. 780–787, 1999.

[135] A. Janshoff, M. Neitzert, Y. Oberdörfer, and H. Fuchs, "Force spectroscopy of molecular systems - single molecule spectroscopy of polymers and biomolecules", *Angew. Chem. Int. Ed.*, vol. 39, pp. 3212–3237, 2000.

[136] N. Abu-Lail and T. Camesano, "Polysaccharide properties probed with atomic force microscopy", *J Microsc*, vol. 212, pp. 217–238, 2003.

[137] J. J. Heinisch, V. Dupres, D. Alsteens, and Y. F. Dufrêne, "Measurement of the mechanical behavior of yeast membrane sensors using single-molecule atomic force microscopy", *Nat. Protoc.*, vol. 5, pp. 670–677, 2010.

[138] G. Francius, D. Alsteens, V. Dupres, S. Lebeer, S. D. Keersmaecker, J. Vanderleyden, H. J. Gruber, and Y. F. Dufrêne, "Stretching polysaccharides on live cells using single molecule force spectroscopy", *Nat. Protoc.*, vol. 4, pp. 939–946, 2009.

[139] D. Alsteens, V. Dupres, S. A. Klotz, N. K. Gaur, P. N. Lipke, and Y. F. Dufrêne, "Unfolding Individual Als5p Adhesion Proteins on Live Cells", *ACS Nano*, vol. 3, pp. 1677–1682, 2009.

[140] F. Ribas, J. Perramon, A. Terradillos, J. Frias, and F. Lucena, "The Pseudomonas group as an indicator of potential regrowth in water distribution systems", *J. Appl. Bacteriol*, vol. 88, pp. 704–710, 2000.

[141] G. O'Toole and R. Kolter, "Initiation of biofilm formation in Pseudomonas fluorescens WCS365 proceeds via multiple, convergent signalling pathways : a genetic analysis", *Mol. Microbiol.*, vol. 28, pp. 449–461, 1998.

[142] D. Allison, B. Ruiz, C. SanJose, A. Jaspe, and P. Gilbert, "Extracellular products as mediators of the formation and detachment of Pseudomonas fluorescens biofilms", *FEMS Microbiol. Lett.*, vol. 167, pp. 179–184, 1998.

[143] R. Lévy and M. Maaloum, "Measuring the spring constant of atomic force microscope cantilevers : thermal fluctuations and other methods", *Nanotechnology*, vol. 13, pp. 33–37, 2002.

[144] C. Novotny, J. Carnahan, and C. C. Brinton, "Mechanical removal of F *pili* type-I *pili* and *flagella* from Hfr and RTF donor cells and kinetics of their reappearance", *J. Bacteriol.*, vol. 98, pp. 1294–1306, 1969.

[145] V. Vadillo-Rodríguez, H. J. Busscher, W. Norde, J. de Vries, R. J. B. Dijkstra, I. Stokroos, and H. C. van der Mei, "Comparision of atomic force microscopy interaction forces between bacteria and silicon nitride substrata for three commonly used immobilization methods", *Appl. Environ. Microbiol.*, vol. 70, pp. 5441–5446, 2004.

[146] A. Touhami, B. Hoffmann, A. Vasella, F. A. Denis, and Y. F. Dufrêne, "Probing specific lectin-carbohydrate interactions using atomic force microscopy imaging and force measurements", *Langmuir*, vol. 19, pp. 1745–1751, 2003.

[147] A. Méndez-Vilas, A. M. Gallardo-Moreno, and M. L. González-Martín, "Nano-mechanical exploration of the surface and sub surface of hydrated cells of *Staphylococcus epidermidis*", *Anton Leew Int J G*, vol. 89, pp. 373–386, 2006.

[148] L. Zhao, D. Schaefer, H. Xu, S. Modi, W. LaCourse, and M. Marten, "Elastic properties of the cell wall of Aspergillus nidulans studied with atomic force microscopy", *Biotechnol. Prog.*, vol. 21, pp. 292–299, 2005.

[149] P. Schar-Zammaretti and J. Ubbink, "The cell wall of lactic acid bacteria : Surface constituents and macromolecular conformations", *Biophys. J.*, vol. 85, pp. 4076–4092, 2003.

[150] S. D. Bentley, D. M. Aanensen, A. Mavroidi, D. Saunders, a. M. C. Ester Rabbinowitsch, K. Donohoe, D. Harris, L. Murphy, M. A. Quail, G. Samuel, I. C. Skovsted, M. S. Kaltoft, B. Barrell, P. R. Reeves, J. Parkhill, and B. G. Spratt, "Genetic analysis of the capsular biosynthetic locus from all 90 pneumococcal serotypes", *PLoS Genet.*, vol. 2, pp. 262–269, 2006.

[151] C. Landersjö, Z. Yang, E. Huttunen, and G. Widmalm, "Structural studies of the exopolysaccharide produced by *Lactobacillus rhamnosus* strain GG (ATCC 53103)", *Biomacromolecules*, vol. 3, pp. 880–884, 2002.

[152] E. Meléndez-Hevia, T. G. Waddell, and E. D. Shelton, "Optimization of molecular design in the evolution of metabolism : the glycogen molecule", *J. Biochem*, vol. 295, pp. 477–483, 1993.

[153] R. Meléndez, E. Meléndez-Hevia, and E. I. Canela, "The fractal structure of glycogen : A clever solution to optimize cell metabolism", *Biophys. J.*, vol. 77, pp. 1327–1332, 1999.

[154] Y. Dufrêne and P. Hinterdorfer, "Recent progress in AFM molecular recognition studies", *Pflugers Arch*, vol. 456, pp. 237–245, 2008.

[155] N. Saitô, K. Takahashi, and Y. Yunoki, "Statistical Mechanical Theory of Stiff Chains", *J. Phys. Soc. Jpn.*, vol. 22, pp. 219–226, 1967.

[156] W. Wang, K. A. Kistler, K. Sadeghipour, and G. Baran, "Molecular dynamics simulation of AFM studies of a single polymer chain", *Phys. Lett. A*, vol. 372, pp. 7007–7010, 2008.

[157] M. Kuhn, H. Janovjak, M. Hubain, and D. Müller, "Automated alignment and pattern recognition of single-molecule force spectroscopy data", *J. Microsc.-Oxf.*, vol. 218, pp. 125–132, 2005.

[158] M. Sullivan, F. Vilaplana, R. Cave, D. Stapleton, A. Gray-Weale, and R. Gilbert, "Nature of alpha and beta Particles in Glycogen Using Molecular Size Distributions", *Biomacromolecules*, vol. 11, pp. 1094–1100, 2010.

[159] G. A. Morris, S. Ang, S. E. Hill, S. Lewis, B. Schäfer, U. Nobbmann, and S. E. Harding, "Molar mass and solution conformation of branched $\alpha(1 \rightarrow 4)$, $\alpha(1 \rightarrow 6)$ Glucans. Part I : Glycogens in water", *Carbohydr. Polym.*, vol. 71, pp. 101–108, 2008.

[160] P. D. O'Grady, B. A. Pearlmutter, and S. T. Rickard, "Survey of sparse and non-sparse methods in source separation", *Int. J. Imag. Syst. Tech., special issue on Blind Source Separation and De-convolution in Imaging and Image Processing*, vol. 15, no. 1, pp. 18–33, 2005.

[161] C. E. Cherry, "Some experiments on the recognition of speech, with one and with two ears", *Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[162] S. Moussaoui, *Séparation de sources non-négatives. Application au traitement des signaux de spectroscopie*, PhD thesis, Université Henri Poincaré, CRAN, Nancy, Dec. 2005.

[163] P. Comon, "Independent component analysis, a new concept ?", *Signal Process.*, vol. 36, no. 3, pp. 287–314, 1994.

[164] A. Hyvärinen, "Survey on independent component analysis", *Neural Computing Surveys*, vol. 2, pp. 94–128, 1999.

[165] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non gaussian signals", *IEE Proceedings-F*, vol. 140, pp. 362–370, 1993.

[166] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution", *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[167] R. Gribonval and S. Lesage, "A survey of sparse component analysis for blind source separation : principles, perspectives, and new challenges", in *ESANN*, 2006, pp. 323–330.

[168] P. Comon and C. Jutten, *Handbook of Blind Source Separation*, Academic Press, 2010.

[169] S. Makino, H. Sawada, R. Mukai, and S. Araki, "Blind source separation of convolutive mixtures of speech in frequency domain", *IEICE Transactions*, vol. 88-A, no. 7, pp. 1640–1655, 2005.

[170] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization.", *Nature*, vol. 401, no. 6755, pp. 788–791, October 1999.

[171] P. O. Hoyer, "Non-negative sparse coding", in *In Neural Networks for Signal Processing XII (Proc. IEEE Workshop on Neural Networks for Signal Processing)*, 2002, pp. 557–565.

[172] K. Torkkola, "Blind separation of convolved sources based on information maximization", in *In IEEE Workshop on Neural Networks for Signal Processing*, 1996, pp. 423–432.

[173] P. Bofill and M. Zibulevsky, "Underdetermined blind source separation using sparse representations", *Signal Processing*, vol. 81, no. 11, pp. 2353–2362, 2001.

[174] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking", *IEEE Trans. Signal Processing*, vol. 52, no. 7, pp. 1830–1847, July 2004.

[175] R. Lambert, *Multichannel blind deconvolution : FIR matrix algebra and separation of multipath mixtures*, PhD thesis, University of Southern California, Faculty of the Graduate School, Department of Electrical Engineering, May 1996.

[176] M. S. Pedersen, J. Larsen, U. Kjems, and L. C. Parra, "A survey of convolutive blind source separation methods", in *Springer Handbook of Speech Processing*. Springer Press, Nov. 2007.

[177] M. G. Jafari, E. Vincent, S. A. Abdallah, M. D. Plumbley, and M. E. Davies, "An adaptive stereo basis method for convolutive blind audio source separation", *Neurocomputing*, vol. 71, no. 10-12, pp. 2087–2097, 2008.

[178] S. Moussaoui, H. Hauksdóttir, F. Schmidt, C. Jutten, J. Chanussot, D. Brie, Douté, and J. A. Benediktsson, "On the decomposition of Mars hyperspectral data by ICA and Bayesian positive source separation", *Neurocomputing*, vol. 71, no. 10-12, pp. 2194–2208, 2008.

[179] N. Bali and A. Mohammad-Djafari, "Bayesian approach with hidden markov modeling and mean field approximation for hyperspectral data analysis", *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 217–225, 2008.

[180] N. Cho and C.-C. J. Kuo, "Underdetermined audio source separation from anechoic mixtures with long time delay", in *Proc. IEEE ICASSP*, 2009, pp. 1557–1560.

[181] R. Saab, Ö. Yilmaz, M. J. McKeown, and R. Abugharbieh, "Underdetermined anechoic blind source separation via ℓ^q-basis-pursuit with $q < 1$", *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4004–4017, 2007.

[182] P. Dierckx, *Curve and surface fitting with splines*, Monographs on Numerical Analysis. Oxford University Press, Inc., New York, NY, May 1995.

[183] A. Hyvärinen and E. Oja, "Independent component analysis : Algorithms and applications", *Neural Networks*, vol. 13, pp. 411–430, 2000.

[184] S. Verdu, *Multiuser Detection*, Cambridge University Press, 1998.

Résumé

Cette thèse s'inscrit dans le domaine des problèmes inverses en traitement du signal. Elle est consacrée à la conception d'algorithmes de restauration et de séparation de signaux parcimonieux et à leur application à l'approximation de courbes de forces en microscopie de force atomique (AFM), où la notion de parcimonie est liée au nombre de points de discontinuité dans le signal (sauts, changements de pente, changements de courbure).

Du point de vue méthodologique, des algorithmes sous-optimaux sont proposés pour le problème de l'approximation parcimonieuse basée sur la pseudo-norme ℓ_0 : l'algorithme Single Best Replacement (SBR) est un algorithme itératif de type "ajout-retrait" inspiré d'algorithmes existants pour la restauration de signaux Bernoulli-gaussiens. L'algorithme Continuation Single Best Replacement (CSBR) est un algorithme permettant de fournir des approximations à des degrés de parcimonie variables. Nous proposons aussi un algorithme de séparation de sources parcimonieuses à partir de mélanges avec retards, basé sur l'application préalable de l'algorithme CSBR sur chacun des mélanges, puis sur une procédure d'appariement des pics présents dans les différents mélanges.

La microscopie de force atomique est une technologie récente permettant de mesurer des forces d'interaction entre nano-objets. L'analyse de courbes de forces repose sur des modèles paramétriques par morceaux. Nous proposons un algorithme permettant de détecter les régions d'intérêt (les morceaux) où chaque modèle s'applique puis d'estimer par moindres carrés les paramètres physiques (élasticité, force d'adhésion, topographie, *etc.*) dans chaque région. Nous proposons finalement une autre approche qui modélise une courbe de force comme un mélange de signaux sources parcimonieux retardées. La recherche des signaux sources dans une image force-volume s'effectue à partir d'un grand nombre de mélanges car il y autant de mélanges que de pixels dans l'image.

Mots-clé

Approximation parcimonieuse, pseudo-norme ℓ_0, Orthogonal Least Squares (OLS), algorithmes de type ajout-retrait, continuation, discontinuités, séparation de sources parcimonieuses, mélanges avec retard, microscopie de force atomique (AFM), imagerie force-volume, modèles physiques de courbes de force.

Abstract

This thesis handles several inverse problems occurring in sparse signal processing. The main contributions include the conception of algorithms dedicated to the restoration and the separation of sparse signals, and their application to force curve approximation in Atomic Force Microscopy (AFM), where the notion of sparsity is related to the number of discontinuity points in the signal (jumps, change of slope, change of curvature).

In the signal processing viewpoint, we propose sub-optimal algorithms dedicated to the sparse signal approximation problem based on the ℓ_0 pseudo-norm : the Single Best Replacement algorithm (SBR) is an iterative "forward-backward" algorithm inspired from existing Bernoulli-Gaussian signal restoration algorithms. The Continuation Single Best Replacement algorithm (CSBR) is an extension providing approximations at various sparsity levels. We also address the problem of sparse source separation from delayed mixtures. The proposed algorithm is based on the prior application of CSBR on every mixture followed by a matching procedure which attributes a label for each peak occurring in each mixture.

Atomic Force Microscopy (AFM) is a recent technology enabling to measure interaction forces between nano-objects. The force-curve analysis relies on piecewise parametric models. We address the detection of the regions of interest (the pieces) where each model holds and the subsequent estimation of physical parameters (elasticity, adhesion forces, topography, *etc.*) in each region by least-squares optimization. We finally propose an alternative approach in which a force curve is modeled as a mixture of delayed sparse sources. The research of the source signals and the delays from a force-volume image is done based on a large number of mixtures since there are as many mixtures as the number of image pixels.

Keywords

Sparse approximation, ℓ_0 pseudo-norm, Orthogonal Least Squares (OLS), forward-backward greedy algorithms, continuation, discontinuities, sparse source separation, delayed mixtures, Atomic Force Microscopy (AFM), force-volume imaging, physical models for force curves.