Gaussian Process Modelling under Inequality Constraints

Andrés F. López-Lopera

N° d'ordre NNT : 2019LYSEM020

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
L'École des Mines de Saint-Étienne

École Doctorale N° 488
Sciences, Ingénierie, Santé

Spécialité de doctorat : Mathématiques Appliquées
Discipline : Science des Données

Soutenue publiquement le 19/09/2019, par:

Andrés F. López-Lopera

Gaussian Process Modelling under Inequality Constraints

Modélisation par Processus Gaussiens sous Contraintes d'Inégalité

Devant le jury composé de :

Prieur, Clémentine	*Professeur, Université Grenoble Alpes, France*	*Présidente*
Kuhnt, Sonja	*Professeur, FH Dortmund, Allemagne*	*Rapportrice*
Nouy, Anthony	*Professeur, École Centrale de Nantes, France*	*Rapporteur*
Filippone, Maurizio	*Professeur Associé (HDR), EURECOM, France*	*Examinateur*
Roustant, Olivier	*Professeur, Mines Saint-Étienne, France*	*Directeur*
Bachoc, François	*Maître de Conférences (HDR), Univ. Paul Sabatier, France*	*Co-encadrant*
Durrande, Nicolas	*Chercheur, PROWLER.io, Angleterre*	*Co-encadrant*

Spécialités doctorales	Responsables :	Spécialités doctorales	Responsables
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche	MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
MECANIQUE ET INGENIERIE	S. Drapier, professeur	INFORMATIQUE	O. Boissier, Professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche	SCIENCES DES IMAGES ET DES FORMES	JC. Pinoli, Professeur
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche	GENIE INDUSTRIEL	N. Absi, Maitre de recherche
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche	MICROELECTRONIQUE	Ph. Lalevée, Professeur

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

ABSI	Nabil	MR	Génie industriel	CMP
AUGUSTO	Vincent	CR	Image, Vision, Signal	CIS
AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BADEL	Pierre	MA(MDC)	Mécanique et ingénierie	CIS
BALBO	Flavien	PR2	Informatique	FAYOL
BASSEREAU	Jean-François	PR	Sciences et génie des matériaux	SMS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BEIGBEDER	Michel	MA(MDC)	Informatique	FAYOL
BLAYAC	Sylvain	MA(MDC)	Microélectronique	CMP
BOISSIER	Olivier	PR1	Informatique	FAYOL
BONNEFOY	Olivier	PR	Génie des Procédés	SPIN
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
CAMEIRAO	Ana	MA(MDC)	Génie des Procédés	SPIN
CHRISTIEN	Frédéric	PR	Science et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	MR	Sciences des Images et des Formes	SPIN
DEGEORGE	Jean-Michel	MA(MDC)	Génie industriel	Fayol
DELAFOSSE	David	PR0	Sciences et génie des matériaux	SMS
DELORME	Xavier	MA(MDC)	Génie industriel	FAYOL
DESRAYAUD	Christophe	PR1	Mécanique et ingénierie	SMS
DJENIZIAN	Thierry	PR	Science et génie des matériaux	CMP
BERGER-DOUCE	Sandrine	PR1	Sciences de gestion	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
DUTERTRE	Jean-Max	MA(MDC)		CMP
EL MRABET	Nadia	MA(MDC)		CMP
FAUCHEU	Jenny	MA(MDC)	Sciences et génie des matériaux	SMS
FAVERGEON	Loïc	CR	Génie des Procédés	SPIN
FEILLET	Dominique	PR1	Génie Industriel	CMP
FOREST	Valérie	MA(MDC)	Génie des Procédés	CIS
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Sciences de la Terre	SPIN
GAVET	Yann	MA(MDC)	Sciences des Images et des Formes	SPIN
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GONDRAN	Natacha	MA(MDC)	Sciences et génie de l'environnement	FAYOL
GONZALEZ FELIU	Jesus	MA(MDC)	Sciences économiques	FAYOL
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
HAN	Woo-Suck	MR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NAVARRO	Laurent	CR		CIS
NEUBERT	Gilles			FAYOL
NIKOLOVSKI	Jean-Pierre	Ingénieur de recherche	Mécanique et ingénierie	CMP
NORTIER	Patrice	PR1	Génie des Procédés	SPIN
O CONNOR	Rodney Philip	MA(MDC)	Microélectronique	CMP
PICARD	Gauthier	MA(MDC)	Informatique	FAYOL
PINOLI	Jean Charles	PR0	Sciences des Images et des Formes	SPIN
POURCHEZ	Jérémy	MR	Génie des Procédés	CIS
ROUSSY	Agnès	MA(MDC)	Microélectronique	CMP
ROUSTANT	Olivier	MA(MDC)	Mathématiques appliquées	FAYOL
SANAUR	Sébastien	MA(MDC)	Microélectronique	CMP
SERRIS	Eric	IRD		FAYOL
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	PR2	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	DR	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR0	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP

Mise à jour : 30/08/2018

Supervisor
Olivier Roustant, Professor, École des Mines Saint-Étienne, France

Co-supervisor
François Bachoc, Senior Lecturer, Université Paul Sabatier, France
Nicolas Durrande, Head of the Probabilistic Modelling Team, PROWLER.io, UK

Jury
Sonja Kuhnt, Professor, FH Dortmund, Germany
Anthony Nouy, Professor, École Centrale de Nantes, France
Clémentine Prieur, Professor, Université Grenoble Alpes, France
Maurizio Filippone, Associate Professor, EURECOM, France

Opponent
Andrés F. López-Lopera

Contact information

Department of Mathematics and Industrial Engineering
Henri Fayol Institute, Mines Saint-Étienne
158 cours Fauriel, F-42023 Saint-Étienne cedex 2, France

Email address: webmaster@emse.fr
URL: https://www.mines-stetienne.fr/
Telephone: +33 (0)4 77 42 01 23
Fax: +33 (0)4 77 42 00 00

This thesis is dedicated to my family:

Carmen Elena and Libardo.

Modélisation par Processus Gaussiens sous Contraintes d'Inégalité

Andrés F. López-Lopera

Génie Mathématique et Industriel
Institut Fayol, Mines Saint-Étienne
158 cours Fauriel, F-42023 Saint-Étienne cedex 2, France
andres-felipe.lopez@emse.fr

Introduction

Les processus gaussiens (PG) représentent un des processus stochastiques les plus communs utilisés dans les cadres bayésiens non paramétriques. Les modèles par PG peuvent être vus comme des distributions sur des espaces de fonctions où des hypothèses (par exemple, la régularité, la stationnarité, et/ou la parcimonie) sont encodées dans des fonctions de covariance (Paciorek and Schervish, 2004; Rasmussen and Williams, 2005; Snelson and Ghahramani, 2006). Le propriétés des PG ont été explorées dans de nombreux processus de décision lié s à des problèmes de régression et de classification (Nickisch and Rasmussen, 2008; Rasmussen and Williams, 2005). La science des données, l'ingénierie, la physique, la biologie et les neurosciences sont des domaines dans lesquels les PG ont été appliqués avec succès (Murphy, 2012; Rasmussen and Williams, 2005).

Cependant, malgré la fiabilité des PG, ces derniers offrent des incertitudes moins réalistes lorsque les systèmes physiques satisfont des contraintes d'inégalité (e.g. conditions de positivité, de monotonie ou de convexité) (Da Veiga and Marrel, 2012; Golchi et al., 2015; Maatouk and Bay, 2017; Zhou et al., 2019). Quantifier proprement ces incertitudes est crucial pour comprendre des phénomènes concrets. Par exemple, dans l'évaluation de la sûreté nucléaire, les environnements expérimentaux exigent généralement des procédures coûteuses et risquées pour évaluer la production de neutrons. Par conséquent, les émulateurs sont nécessaires pour l'inférence de ces productions et doivent intégrer les contraintes d'une sortie à la fois positive et monotone par rapport à un ensemble de paramètres d'entrée. Pour obtenir des prévisions plus précises, ces deux conditions doivent être prises en compte dans la quantification de l'incertitude. D'autres cas de test où les données montrent des contraintes d'inégalité spécifiques se rencontrent dans les réseaux informatiques (monotonie) (Golchi et al., 2015), l'analyse du système social (monotonie) (Riihimäki and Vehtari, 2010), l'économétrie (monotonie ou positivité) (Cousin et al., 2016), et la physique nucléaire (monotonie et/ou convexité) (Zhou et al., 2019).

Plusieurs études ont montré que l'inclusion de contraintes d'inégalité dans les cadres de PG peut conduire à une quantification plus réaliste de l'incertitude lors de l'apprentissage de données réelles (Da Veiga and Marrel, 2012; Golchi et al., 2015; Riihimäki and Vehtari,

2010; Zhang and Lin, 2018). Dans la plupart des cas, on suppose que les inégalités sont satisfaites sur un ensemble fini d'emplacements en entrée. Ensuite, la distribution postérieure est approchée compte tenu de ces entrées contraintes (voir, par exemple, Golchi et al., 2015; Riihimäki and Vehtari, 2010). En pratique, une alternative pour traiter les contraintes de positivité, de monotonie ou de convexité consiste à utiliser des intégrales (itérées) de processus positifs (par exemple, échelle logarithmique des PG, Vanhatalo and Vehtari, 2007). Cependant, ces approches ont une densité avec une masse nulle à zéro et sont limitées à des conditions d'inégalité spécifiques. Dans (Zhang and Lin, 2018), une nouvelle projection de PG est développée afin d'incorporer des conditions de borne sur tout le domaine d'entrée. Bien qu'une solution de forme fermée pour la projection y soit fournie, leur cadre est limité à la prise en compte des contraintes de borne. A la connaissance générale, l'approche de Maatouk and Bay (2017) est la seule approche gaussienne proposée dans la littérature qui satisfait des inégalités spécifiques et en particulier l'espace d'entrée. Là, les échantillons par GP sont approximés dans des espaces de dimensions finies telles que des fonctions linéaires par morceaux. Bay et al. (2016) ont montré que le mode postérieur du processus sous les contraintes converge vers celui fourni par l'interpolation par splines. Cette approche a été appliquée à plusieurs types de données réelles : l'économétrie (Cousin et al., 2016), la géostatistique (Maatouk and Bay, 2017), et la physique nucléaire (Zhou et al., 2019).

Le cadre proposé par Maatouk and Bay (2017) présente cependant quelques limitations. Premièrement, même s'il s'agit d'un ensemble convexe général d'inégalités linéaires, il est concentré sur l'interpolation des données sous conditions de bornes, monotonie ou convexité. Deuxièmement, la méthode d'échantillonnage de rejet proposée dans (Maatouk and Bay, 2016) pour estimer la distribution postérieure entraîne un taux de rejet élevé lorsque l'ordre d'approximation finie augmente, ou que les contraintes d'inégalité deviennent plus complexes. De plus, leur modèle permet des implémentations allant jusqu'à des domaines d'entrée bidimensionnels (en conséquence de l'inconvénient précédent). Enfin, la technique de validation croisée proposée par Maatouk et al. (2015) pour l'estimation des paramètres, limite les valeurs optimales à une grille finie de valeurs possibles et fournit la même estimation des paramètres de corrélation que pour les PG sans contrainte. Afin de remédier à ces limitations, les contributions de cette thèse sont les suivantes. Premièrement, nous étudions plus en détail une approximation en dimension finie pouvant prendre en compte des contraintes générales d'inégalité linéaires. Deuxièmement, nous étudions l'emploi de méthodes de Monte Carlo et Monte Carlo par chaînes de Markov pour l'approximation de la distribution postérieure du modèle. Troisièmement, en ce qui concerne l'extension pour de grandes dimensions (impliquant des dizaines ou centaines de variables d'entrée, par exemple), nous explorons différentes constructions de l'approximation finie proposée. Nous remarquons que l'introduction d'un bruit d'observation permet de monter à la dimension cinq. Nous proposons un algorithme d'insertion des nœuds, qui concentre le budget de calcul sur les dimensions les plus actives. Nous explorons aussi la triangulation de Delaunay comme alternative à la tensorisation. Enfin, nous étudions l'utilisation de modèles additifs dans ce contexte, théoriquement et expérimentalement sur des problèmes de plusieurs centaines de

variables. Quatrièmement, nous donnons des résultats théoriques sur l'inférence sous contraintes d'inégalité. La consistance et la normalité asymptotique d'estimateurs par maximum de vraisemblance sont établies. Finalement, nous montrons que les implémentations de ce manuscrit peuvent être couplées à d'autres types de processus modulés par PG, où des priors gaussiens sous contraintes d'inégalité sont nécessaires pour obtenir des modèles aussi précis que possible avec des incertitudes plus réalistes (e.g. dans le cadre des processus de Cox et des processus de renouvellement).

Il faut noter que des contributions récentes fondées sur le cadre proposé par Maatouk (2015) ont été suggérées lors de la préparation de ce manuscrit (voir, par exemple, Maatouk, 2017; Ray et al., 2019; Zhou et al., 2019). Dans (Maatouk, 2017), le modèle par PG dans (Maatouk, 2015) a été adapté pour rendre compte de l'observation bruitée sous des contraintes de borne, monotonie ou convexité. D'autres travaux de Zhou et al. (2019) et Ray et al. (2019) ont permis la combinaison de contraintes (par exemple, monotonicité et convexité). Contrairement à ces contributions, les développements effectués durant cette thèse tiennent compte des contraintes générales d'inégalité linéaires et ils ne nécessitent pas de calcul des dérivées des fonctions de covariance. Ceci implique qu'il n'y a aucune restriction sur le choix de la fonction de covariance (e.g. en ce qui concerne la monotonie ou la convexité). Nous n'avons pas été en mesure de comparer notre cadre à ceux proposés dans (Ray et al., 2019; Zhou et al., 2019) au cours de la rédaction de ce manuscrit. Cependant, nous savons que ces approches existent et, par conséquent, les comparaisons numériques présentent un intérêt pour les perspectives à venir.

Résumé des Chapitres

Dans la Partie II, un aperçu des bases des modèles de régression par PG est exposé. Dans le Chapitre 2, nous commençons par le point de vue classique, définissant les termes clés des PG (fonction de covariance et estimateur de vraisemblance maximale, par exemple). Ensuite, nous expliquerons brièvement comment adapter les modèles par PG lorsque des fonctions additives sont considérées.

Dans la Partie III, le cadre proposé par Maatouk (2015) est étudié plus en détail afin de prendre en compte des contraintes générales d'inégalité linéaires (Chapitre 3). Cela conduit à des modèles plus polyvalents pouvant être utilisés pour une large gamme d'applications. En outre, l'approche de cette thèse ne comporte aucune restriction quant au choix de la fonction de covariance (par exemple, par rapport à la monotonie ou la convexité). Le cadre résulte en une distribution postérieure gaussienne tronquée qui peut être approximée par des méthodes de Monte Carlo (MC) et Monte Carlo par chaînes de Markov (MCCM). Ainsi, pour rendre le modèle développé applicable aux implémentations concrètes, des échantillonneurs par MC et MCCM ont été explorés. Cette approche a ensuite finalement été testée sur une application nucléaire en dimension deux, où les contraintes de borne et monotonicité sont satisfaites.

Les observations bruitées sont prises en compte dans le Chapitre 4. La relaxation des contraintes d'interpolation par un effet de bruit entraîne des espaces d'échantillonnage

moins restrictifs dans lesquels sont effectués des échantillonneurs par MC et MCCM. Cela conduit à des implémentations plus rapides tout en préservant des taux d'échantillonnage effectifs élevés. Il a été démontré sur divers exemples synthétiques, et sans hypothèses supplémentaires, que le cadre établi pendant la thèse est applicable en dimension cinq, et ce, pour des milliers d'observations. Pour finir, le modèle développé a été évalué sur des applications d'inondations côtières en dimension un et cinq.

Dans la Partie IV, en ce qui concerne l'extension en grandes dimensions, des constructions alternatives du cadre proposé dans la Partie III sont explorées. Premièrement, en raison de la construction du tenseur dans les Chapitres 3 et 4, les implémentations deviennent coûteuses avec une résolution plus fine dans l'approximation. Par conséquent, il est préférable d'augmenter la qualité de la représentation uniquement dans les régions plus variables. Pour ce faire, dans le Chapitre 5, un algorithme séquentiel a été exploré pour la construction automatique de grilles rectangulaires (non équidistantes) des nœuds utilisés dans la tensorisation. Les performances de l'algorithme proposé ont été testées sur des exemples synthétiques allant jusqu'à la dimension cinq.

Dans le Chapitre 6, le cadre par PG est adapté aux grandes dimensions en étudiant le cas où des fonctions satisfont des conditions additives ou additives par blocs. Puisque les contraintes sont supposées être imposées sur un sous-ensemble prédéfini de variables d'entrée, les développements proposés dans les Chapitres 3, 4 et 5 peuvent être appliqués efficacement à des sous-espaces (généralement) de dimension faible. Cela conduit à des modèles par PG sous contraintes qui peuvent facilement être mis à l'échelle dans des grandes dimensions impliquant des centaines de variables d'entrée.

La Partie V s'intéresse à l'inférence sous contraintes d'inégalité. La vraisemblance sous contraintes est obtenue en conditionnant celle qui n'est pas des contraintes au fait que les inégalités sont satisfaites (Chapitre 7). Il est montré que, de manière générale, tout résultat de consistance pour MV avec des PG sans contraintes est préservé lors de l'ajout de contraintes de borne, monotonie et convexité. Ce chapitre présente également l'étude de certains cas où les estimateurs non-contraints et contraints sont distribués asymptotiquement gaussiens, à condition que le PG réponde aux contraintes. De plus, il a également été montré que l'estimateur MV sous contraintes, appliqué à différents exemples, est généralement plus précis sur des échantillons finis.

Les implémentations décrites dans les chapitres précédents peuvent également être couplées à d'autres types de processus stochastiques modulés par des priors gaussiens : e.g. les processus de Cox (Partie VI). Dans les processus de Cox, la fonction d'intensité est modélisée comme un PG positif. Ainsi, le Chapitre 8 est marqué par l'introduction d'une nouvelle approximation en dimension finie des processus de Cox modulés par PG, dans laquelle des contraintes de positivité peuvent être imposées directement sur le prior gaussien. La polyvalence du cadre résultant pour prendre en compte tout type de contrainte d'inégalité conduit à des modèles pouvant être utilisés pour d'autres classes de processus ponctuels : e.g. pour les processus de renouvellement. Il a été démontré à la fois sur des données synthétiques et réelles, que le cadre développé durant cette thèse donne des résultats d'inférence précis qui sont compétitifs par rapport à ceux fournis par

d'autres méthodes de la littérature.

Les principaux développements de ce manuscrit sont implémentés dans le langage de programmation R et font partie du package `lineqGPR`: modèles de régression par processus gaussiens avec contraintes d'inégalité (Partie VII). Nous y voyons une contribution importante pour une utilisation pratique ainsi qu'un outil de valeur pour la recherche. Dans le Chapitre 9, une brève description de certaines de ses principales fonctionnalités est donnée sous diverses illustrations numériques.

Enfin, dans la Partie VIII, un aperçu des perspectives et un résumé des conclusions sont exposés. En particulier, dans le Chapitre 10, une discussion d'une représentation alternative par PG basée sur d'une triangulation de Delaunay. Le nombre de termes dans ce PG de dimension fini n'augmentons pas de manière exponentielle avec les dimensions, nous conforte dans l'idée que ces implémentations peuvent être appliquées de manière plus efficace en grandes dimensions.

Conclusions

Les développements de ce manuscrit sont en phase avec trois directions principales : 1) l'améliorer de l'applicabilité d'un modèle de type PG sous contraintes sous contraintes d'inégalité en fournissant un échantillonneur rapide, 2) rendre les modèles capables de monter à de grandes dimensions et/ou à un grand nombre d'observations, et 3) l'étudier de l'inférence sous contraintes d'inégalité.

De manière générale, il a été montré que l'inclusion de contraintes d'inégalité dans les cadre de PG permet d'obtenir des modèles plus réalistes. Cette thèse s'est intéressée au modèle de type PG proposé par Maatouk (2015), obtenu par approximation finie, qui garantit que les contraintes sont satisfaites dans tout l'espace. Plus concrètement, plusieurs contributions ont été apportées. Premièrement, nous avons étudié l'emploi de méthodes de Monte Carlo et Monte Carlo par chaînes de Markov pour des lois multinormales tronquées. Elles ont fourni un échantillonnage efficace pour des contraintes d'inégalité linéaires. Deuxièmement, nous avons exploré l'extension du modèle, jusque-là limité à la dimension trois, à de plus grandes dimensions. Il a été remarqué que l'introduction d'un bruit d'observations permet de monter à la dimension cinq. Nous avons proposé un algorithme d'insertion des nœuds, qui concentre le budget de calcul sur les dimensions les plus actives. La triangulation de Delaunay a aussi été explorée comme alternative à la tensorisation. Enfin, nous avons étudié l'utilisation de modèles additifs dans ce contexte, théoriquement et expérimentalement sur des problèmes de plusieurs centaines de variables. Troisièmement, nous avons donné des résultats théoriques sur l'inférence sous contraintes d'inégalité. La consistance et la normalité asymptotique d'estimateurs par maximum de vraisemblance ont été établis. L'ensemble des travaux a fait l'objet d'un développement logiciel en R. Ils ont été appliqués à des problèmes de gestion des risques en sûreté nucléaire et inondations côtières, avec des contraintes de positivité et monotonie. En ouverture, nous avons montré que la méthodologie fournit un cadre original pour l'étude de processus de Poisson d'intensité stochastique.

Table des Matières

Contents

List of Figures

List of Tables

Part I

Preface

Chapter 1

General Aspects

Contents

1.1 Preliminary context

This thesis is conducted within the frame of the Chair in Applied Mathematics OQUAIDO, gathering partners in technological research (BRGM, CEA, IFPEN, IRSN, Safran, Storengy) and academia (CNRS, École Centrale de Lyon, Mines Saint-Étienne, Université Grenoble, Université Nice, Université de Toulouse) around advanced methods for the design and analysis of computer experiments. Its research domain is devoted to investigating industrial computer codes that are costly-to-evaluate. Such computer codes typically aim at modelling complex physical phenomena such as encountered in the development of new technologies or risk studies. They are used by the partners of the Chair in different research fields such as energy, environment and transport problems.

This thesis is inspired by previous scientific contributions associated with the ReDICE consortium. More precisely, it continues the research line investigated by Maatouk (2015) in the thesis entitled: *Correspondence between Gaussian Process Regression and Interpolation Splines under Linear Inequality Constraints: Theory and Applications*. As in (Maatouk, 2015), this work is dedicated to study stochastic interpolation and regression models based on Gaussian processes (GPs) under inequality constraints (e.g. boundedness, monotonicity, convexity). Throughout this manuscript, we overcome some limitations from (Maatouk, 2015) and introduce novel GP developments in three main directions: 1) to improve the applicability of GPs accounting for inequality constraints by providing a fast sampler, 2) to make the constrained GP models scalable to higher dimensions and/or number of observations, and 3) to investigate estimation under inequality constraints.

1.2 Introduction

GPs are one of the most famous stochastic processes used in non-parametric Bayesian frameworks. GP models place prior distributions over function spaces, and prior assumptions (e.g. smoothness, stationarity, sparsity) are encoded in the covariance function (Paciorek and Schervish, 2004; Rasmussen and Williams, 2005; Snelson and Ghahramani, 2006). The properties of GPs have been explored in many decision tasks in both regression and classification problems (Nickisch and Rasmussen, 2008; Rasmussen and Williams, 2005). Computer science, physics, biology, and neuroscience are some fields where GPs have been applied successfully (Murphy, 2012; Rasmussen and Williams, 2005).

Despite the reliable performance of GPs, they are likely to provide unrealistic uncertainty estimates when physical systems satisfy inequality constraints (e.g. boundedness, monotonicity or convexity) (Da Veiga and Marrel, 2012; Golchi et al., 2015; Maatouk and Bay, 2017; Zhou et al., 2019). Quantifying properly those uncertainties is crucial for understanding real-world phenomena. For example, in nuclear safety criticality assessment, experimental settings typically require expensive and risky procedures to evaluate neutron productions. Hence, emulators are required to infer these production rates and should encode the constraint that the output is positive and monotonic with respect to a given set of input parameters. In this sense, in order to obtain more accurate predictions, both conditions have to be considered in the uncertainty quantification. Other test cases where data exhibit specific inequality constraints are given in computer networking (monotonicity) (Golchi et al., 2015), social system analysis (monotonicity) (Riihimäki and Vehtari, 2010), econometrics (monotonicity or positivity) (Cousin et al., 2016), and nuclear physics (monotonicity and/or convexity) (Zhou et al., 2019).

Several studies have shown that including inequality constraints in GP frameworks can lead to more realistic uncertainty quantifications in learning from real data (Da Veiga and Marrel, 2012; Golchi et al., 2015; Riihimäki and Vehtari, 2010; Zhang and Lin, 2018). In most cases, it is assumed that the inequalities are satisfied on a finite set of input locations. Then, the posterior distribution is approximated given those constrained inputs (see, e.g., Golchi et al., 2015; Riihimäki and Vehtari, 2010). In practice, an alternative to deal either with positiveness, monotonicity, or convexity constraints is to use (iterated) integrals of positive processes (e.g. log-GPs, Vanhatalo and Vehtari, 2007). However, those approaches have a density with zero mass in zero and are limited to specific inequality conditions. In (Zhang and Lin, 2018), a novel GP projection is developed aiming at incorporating bound information along the input domain. Although a closed-formed solution for the projection is provided there, their framework is limited to account only for boundedness constraints. In (Maatouk and Bay, 2017), GP samples are approximated in finite-dimensional spaces of functions such as piecewise linear functions. It is shown by Bay et al. (2016) that the posterior mode of the resulting constrained process converges to the one provided by spline interpolation. Such GP-based framework in (Maatouk and Bay, 2017) has been applied on several real-data (e.g. econometrics, Cousin et al. (2016), geostatistics, Maatouk and Bay (2017), nuclear physics Zhou et al. (2019)), resulting in more realistic uncertainties than unconstrained GP models.

The framework proposed by Maatouk and Bay (2017) still presents some limitations. First, even if it was written for a general convex set of linear inequalities, the focus was on data interpolation with investigations for either boundedness, monotonicity or convexity conditions. Second, the proposed rejection sampling method in (Maatouk and Bay, 2016) for estimating the posterior results in a high rejection rate when either the order of the finite approximation increases or the inequality constraints become more complex. Third, their model leads to tractable implementations up to two-dimensional input domains as a consequence of the previous drawback. Fourth, the proposed leave-one-out technique for parameter estimation in (Maatouk et al., 2015) restricts the optimal values to be on a finite grid of possible values, and provides the same estimation of correlation parameters as for unconstrained GP parameters. In order to address these limitations, the contributions of this thesis are fivefold. First, we extend the framework to deal with general sets of linear inequality constraints. Second, we evaluate efficient Monte Carlo (MC) and Markov Chain Monte Carlo (MCMC) algorithms that can be used to approximate the posterior distribution. Third, we explore alternative constructions of finite-dimensional GP models and/or consider additional assumptions (e.g. noise effect, additive structures) aiming at scaling models to higher dimensions (i.e. involving tens or hundreds of inputs variables). Fourth, we investigate theoretical and numerical properties of a conditional likelihood for covariance parameter estimation accounting for inequality constraints. Finally, we demonstrate that implementations throughout this manuscript can be coupled to other types of GP-modulated processes where constrained Gaussian priors are crucial to get accurate models with more realistic uncertainties (e.g. Cox processes, renewal processes).

One must note that recent contributions based on the framework proposed by Maatouk (2015) have been suggested during the preparation of this manuscript (see, e.g., Maatouk, 2017; Ray et al., 2019; Zhou et al., 2019). In (Maatouk, 2017), the GP-based model in (Maatouk, 2015) was adapted to account for noisy observation under boundedness, monotonicity or convexity constraints. Further works by Zhou et al. (2019) and (Ray et al., 2019) allowed the combination of constraints (e.g. monotonicity and convexity). In contrast to those contributions, our developments account for general sets of linear inequality constraints and do not require to compute kernel derivatives. The latter attribute implies that there is no restrictions on the choice of the covariance function (e.g. as for monotonicity or convexity). We were not able to compare our framework with respect to the ones proposed in (Ray et al., 2019; Zhou et al., 2019) during the writing process of this manuscript. However, we know that those approaches exist, and therefore, numerical comparisons are of interest in future implementations.

1.3 Structure of the manuscript

The developments throughout this manuscript are in phase with three main directions: 1) to improve the applicability of GPs accounting for inequality constraints by providing a fast sampler, 2) to make the constrained GP models scalable to higher dimensions and/or

number of observations, and 3) to investigate estimation under inequality constraints.

In Part II, we give an overview of the basics on GP regression models. In Chapter 2, we start from the classical point of view, defining key terms of GPs (e.g. covariance function and maximum likelihood estimator). Then, we briefly explain how to adapt GP implementations when additive functions are considered.

In Part III, we further investigate the framework proposed by Maatouk (2015) to account for general linear inequality constraints (see Chapter 3). This leads to more versatile models that can be used for a broad range of applications. Furthermore, our approach does not have restrictions on the choice of the covariance function (e.g. as for monotonicity or convexity). The computations involve truncated Gaussian posterior distributions, that can be approximated via MC and MCMC algorithms. Thus, we explore efficient MC/MCMC samplers to make the model applicable in practice. Finally, we test the proposed approach on a 2D nuclear application where both boundedness and monotonicity constraints are satisfied.

Noisy observations are considered in Chapter 4. The relaxation of the interpolation constraints through a noise effect results in less restrictive sample spaces where MC/MCMC samplers are performed. This leads to faster implementations while preserving high effective sampling rates. We demonstrate on various synthetic examples, and without further assumptions, that the framework is applicable up to 5D and for thousands of observations. Finally, we assess the proposed model on 2D and 5D coastal flooding applications.

In Part IV, regarding the extension in high dimensions, we explore alternative constructions of the framework in Part III. First, due to the tensor construction in Chapters 3 and 4, implementations become costly with finer resolution in the approximation. Therefore, it is worth preferentially increasing the quality of representation only in highly variable regions. To do so, in Chapter 5, we explore a sequential algorithm for the automatic construction of (non-equispaced) rectangular grids of the knots used in the tensorisation. The algorithm is inspired from the free-knots paradigm in spline constructions. We test the performance of the proposed algorithm on synthetic examples up to 5D.

In Chapter 6, we adapt the framework to deal with high dimensions by studying the case where functions satisfy additive or block-additive conditions. Since constraints are assumed to be imposed on a predefined subset of input variables, developments in Chapters 3 to 5 can be efficiently applied on (usually) low-dimensional subspaces. This leads to constrained GP models that can be easily scaled in high dimensions involving hundreds of input variables.

In Part V, we focus on inference under inequality constraints. The constrained likelihood is obtained by conditioning the unconstrained one to the fact that the inequalities are satisfied (see Chapter 7). We show that, loosely speaking, any consistency result for the maximum likelihood estimator (MLE) with unconstrained GPs, is preserved when adding boundedness, monotonicity and convexity constraints. We also study some cases where both the unconstrained and constrained estimators are asymptotically Gaussian distributed, conditionally to the fact that the GP satisfies the constraints. We show in

simulations that the constrained MLE is generally more accurate on finite samples.

The GP implementations in previous chapters can also be coupled to other types of stochastic processes which are modulated by Gaussian priors: e.g. Cox processes (see Part VI). In Cox processes, the intensity function is modelled as a positive GP. Thus, in Chapter 8, we introduce a novel finite-dimensional approximation of GP-modulated Cox processes where positivity constraints can be imposed directly on the GP prior. The versatility of the resulting framework to account for any type of inequality constraints leads to models that can be used for other classes of point processes: e.g. renewal processes. We demonstrate on both synthetic and real-world data that the proposed framework gives accurate inference results that are competitive with those provided by other methods from the state-of-the-art.

The main GP developments throughout this manuscript are implemented in the R programming language, and they are part of the package `lineqGPR`: Gaussian process regression models with inequality constraints (see Part VII). We see it as an important contribution for practical usage as well as a valuable tool for research. In Chapter 9, we briefly describe some of its main functionalities on various numerical illustrations.

Finally, in Part VIII, we give an insight of the perspectives and summarise the conclusions. In particular, in Chapter 10, we discuss an alternative representation of GPs based on Delaunay triangulations. We believe that, since the number of terms in this new finite-dimensional GP does not increase exponentially with the dimensions, GP implementations can be scaled more efficiently to high dimensions.

1.4 Scientific contributions

Results throughout this thesis are based on diverse scientific contributions including publications in international journals, proceedings in international conferences, preprints, conferences, R packages, etc. Next, we list the main scientific contributions.

Publications in international journals

1. López-Lopera, A. F., Bachoc, F., Durrande, N., and Roustant, O. (2018). Finite-dimensional Gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1224–1255.

2. Bachoc, F., Lagnoux, A., and López-Lopera, A. F. (2019). Maximum likelihood estimation for Gaussian processes under inequality constraints. *Electronic Journal of Statistics*, 13(2):2921–2969.

Proceedings in international conferences

3. López-Lopera, A. F., John, S., and Durrande, N. (2019). Gaussian process modulated Cox processes under linear inequality constraints. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, Naha, Okinawa, Japan.

4. López-Lopera, A. F., Bachoc, F., Durrande, N., Rohmer, J., Idier, D., and Roustant, O. (2018). Approximating Gaussian process emulators with linear inequality constraints and noisy observations via MC and MCMC. In *Monte Carlo and Quasi-Monte Carlo Methods (MCQMC)*, Rennes, France.

Conferences & workshops

5. (Poster) López-Lopera, A. F., Bachoc, F., Durrande, N., and Roustant, O. (2018). Gaussian process regression models under linear inequality conditions. In *Mascot-Num*,[1] Nantes, France.

6. López-Lopera, A. F., Bachoc, F., Durrande, N., and Roustant, O. (2018). Finite dimensional Gaussian approximation with linear inequality constraints. In *SIAM Conference on Uncertainty Quantification (SIAM-UQ)*, California, USA.

7. López-Lopera, A. F., Bachoc, F., Durrande, N., and Roustant, O. (2019). Gaussian process regression models under linear inequality conditions. In *Mascot-Num*, Rueil-Malmaison, France.

8. (Poster) López-Lopera, A. F. (2019). lineqGPR: an R package for Gaussian process regression modelling with linear inequality constraints. In *International Conference for Users of R (UseR!)*, Toulouse, France.

R packages

9. López-Lopera, A. F. (2019). lineqGPR: Gaussian Process Regression Models with Inequality Constraints. R package version 0.0.4.

International scientific exchanges

- Scientific visit at PROWLER.io,[2] Cambridge, UK, February, 2018.

Participations to the Chair OQUAIDO

- GP modelling under Inequality Constraints I – Progress on Chapter 3, in Chair OQUAIDO Scientific Days, Université de Nice, France, May 10–11, 2017.

- GP modelling under Inequality Constraints II – Progress on Chapters 3 and 7, in Chair OQUAIDO Scientific Days, BRGM, Orléans, France, Nov. 22–23, 2017.

- Package lineqGPR – Progress on Chapter 9, in Chair OQUAIDO Software Training, Mines Saint-Étienne, France, Jan. 30, 2018.

[1] "Méthodes d'Analyse Stochastique pour les Codes et Traitements Numériques" (in French).

[2] PROWLER.io is a research led, Cambridge based startup focusing on behavioural learning and simulation in virtual environments.

- GP modelling under Inequality Constraints III – Progress on Chapters 7, 8 and 10, in Chair OQUAIDO Scientific Days, Centrale Lyon, France, Jan. 22–24, 2018.

- GP modelling under Inequality Constraints IV – Progress on Chapters 5 and 6, in Chair OQUAIDO Scientific Days, CEA, Cadarache, France, Nov. 22–23, 2018.

- GP modelling under Inequality Constraints V – Coastal flooding applications in Chapter 4, in Chair OQUAIDO Scientific Days, IMT, Toulouse, France, May 14–16, 2019.

Part II

Background

Chapter 2

Basics on Gaussian Process Regression Models

Contents

2.1 Introduction

Since developments along this manuscript are built on standard GP models, this chapter contains a collection of notations and definitions needed in later parts. In Section 2.2, we start from the classical point of view, defining key terms of GP regression models (e.g. covariance function and maximum likelihood estimator). Then, in Section 2.3, we briefly explain how to adapt GP models when additive functions are considered. In contrast to standard GP regression models, the main benefit of considering additivity relies in the fact that an exponentially increasing number of observations is not required to cover the input space.

2.2 Gaussian process regression modelling

In this manuscript, we consider latent functions $f : \mathcal{D} \to \mathbb{R}$ defined on a compact input space $\mathcal{D}$. The definition of $\mathcal{D}$ may vary among the different types of models and applications. It may vary from unit intervals in one-dimensional problems, i.e. $\mathcal{D} = [0, 1]$; to d-dimensional unit hypercubes in high dimensions, i.e. $\mathcal{D} = [0, 1]^d$. We assume that a finite number of observations n are available at points $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathcal{D}$, leading to a vector of observations $\mathbf{f}$ defined as:

$$\mathbf{f} = \left[f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_n)\right]^\top.$$

We consider stochastic processes given by

$$Y : (\Omega, \mathcal{A}) \to (C(\mathcal{D}, \mathbb{R}), \mathcal{B}),$$

where $(\Omega, \mathcal{A})$ is a measurable space, $C(\mathcal{D}, \mathbb{R})$ is the set of continuous functions from $\mathcal{D}$ to $\mathbb{R}$, and $\mathcal{B}$ is the Borel sigma algebra on $C(\mathcal{D}, \mathbb{R})$ corresponding to the L^∞ norm (Stein, 1999).

Before introducing the definition of a GP, concepts of symmetry and positive semi-definiteness are required since they play an important role for the construction of covariance functions (Genton, 2001; Rasmussen and Williams, 2005).

Definition 2.1 (Symmetry) *Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is symmetric if $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$,*

$$k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}', \boldsymbol{x}).$$

Definition 2.2 (Positive semi-definiteness) *A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive semidefinite (p.s.d.) if for all $n \in \mathbb{N}$, and for all $a_1, \cdots, a_n \in \mathbb{R}, \forall\ \boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathcal{X}$,*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0.$$

Definition 2.3 (Covariance function) *Consider a symmetric and p.s.d. function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then, k is said to be a valid covariance function (or kernel) on $\mathcal{X}$.*

Thus, the function k is p.s.d. if for every choice of design points $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$, the matrix $(k(\boldsymbol{x}_i, \boldsymbol{x}_j))_{1 \leq i,j, \leq n}$ is p.s.d.

2.2.1 Gaussian process

A GP is a collection of random variables, such that any finite subset of them have a joint Gaussian distribution (Rasmussen and Williams, 2005). Let $\{Y(\boldsymbol{x}); \boldsymbol{x} \in \mathcal{D}\}$ be a GP. Then, Y is completely defined by its mean function μ and covariance function k:

$$Y \sim \mathcal{GP}(\mu, k), \tag{2.1}$$

Table 2.1: Common stationary kernel functions used in GP regression models in 1D.

Type of kernel	Expression for $k(x, x')$	Class
Squared Exponential (SE)	$\sigma^2 \exp\left\{-\frac{(x-x')^2}{2\ell^2}\right\}$	C^∞
Matérn 5/2	$\sigma^2\left(1+\frac{\sqrt{5}\lvert x-x'\rvert}{\ell}+\frac{5(x-x')^2}{3\ell^2}\right)\exp\left\{-\frac{\sqrt{5}\lvert x-x'\rvert}{\ell}\right\}$	C^2
Matérn 3/2	$\sigma^2\left(1+\frac{\sqrt{3}\lvert x-x'\rvert}{\ell}\right)\exp\left\{-\frac{\sqrt{3}\lvert x-x'\rvert}{\ell}\right\}$	C^1
Exponential	$\sigma^2 \exp\left\{-\frac{\lvert x-x'\rvert}{\ell}\right\}$	C

where $\mu(\boldsymbol{x}) = \mathbb{E}\{Y(\boldsymbol{x})\}$ and $k(\boldsymbol{x}, \boldsymbol{x}') = \operatorname{cov}\{Y(\boldsymbol{x}), Y(\boldsymbol{x}')\}$ for $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{D}$. The operator $\mathbb{E}$ denotes the expectation of random variables, and the covariance operator is given by

$$\operatorname{cov}\{Y(\boldsymbol{x}), Y(\boldsymbol{x}')\} = \mathbb{E}\{[Y(\boldsymbol{x}) - \mu(\boldsymbol{x})][Y(\boldsymbol{x}') - \mu(\boldsymbol{x}')]\}.$$

For ease of notation, we focus on centred GP prior assumptions, i.e. Y has mean function μ equal to zero.[1] For a given finite number of points $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$, then the vector with random variables (r.v.'s) $\mathbf{Y}_n = \left[Y(\boldsymbol{x}_1), \cdots, Y(\boldsymbol{x}_n)\right]^\top$, satisfies

$$\mathbf{Y}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{K}), \tag{2.2}$$

with covariance matrix $\mathbf{K} = (\operatorname{cov}\{Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_j)\})_{1\leq i,j\leq n}$.

2.2.1.1 Kernel functions

Let $\{Y(\boldsymbol{x}); \boldsymbol{x} \in \mathcal{D}\}$ be a centred GP. Since Y has a mean function equal to zero, it can be fully described by its kernel function,

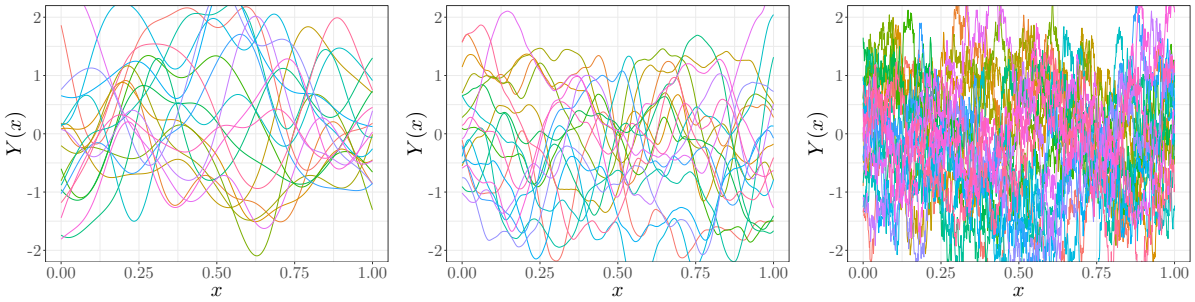
$$k(\boldsymbol{x}, \boldsymbol{x}') = \operatorname{cov}\{Y(\boldsymbol{x}), Y(\boldsymbol{x}')\}.$$

Different choices of prior assumptions can be encoded in Y depending on the choice of the kernel k (e.g. smoothness, stationary). For simplicity and ease of notation, we assume stationarity in the next chapters. However, GP implementations proposed along the manuscript are not restricted to stationarity, and further developments can be made for non-stationary kernels.

Definition 2.4 (Stationary kernel functions) *A kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, with $\mathcal{X} \subset \mathbb{R}^d$, is stationary if, for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, $k(\boldsymbol{x}, \boldsymbol{x}')$ only depends on $\boldsymbol{x} - \boldsymbol{x}'$.*

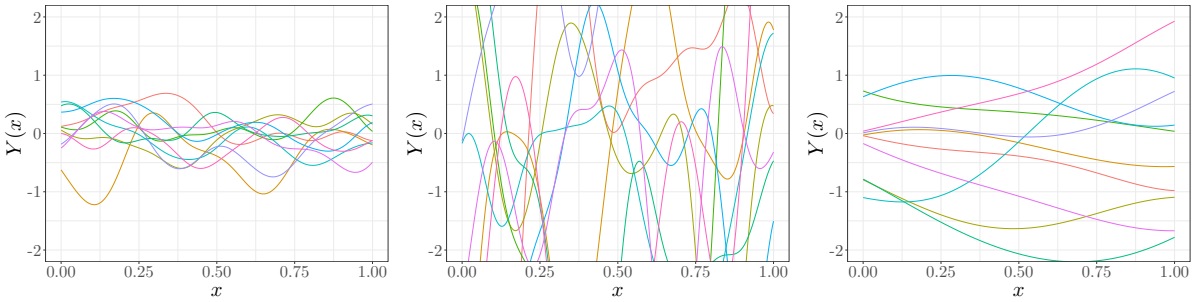
Table 2.1 shows the most common stationary kernels used in GP regression models in 1D. They are shown in decreasing order of regularity in mean square sense (Santner et al., 2003a; Stein, 1999), where the class C^∞ corresponds to the space of infinitely differentiable

[1] A GP Z with mean function μ and kernel k can be expressed in terms of a centred GP Y with same kernel function: $Z(\boldsymbol{x}) = \mu(\boldsymbol{x}) + Y(\boldsymbol{x})$.



(a) SE (b) Matérn 5/2 (c) Exponential

Figure 2.1: Effect of different kernels from Table 2.1 on unconditional GP samples with fixed covariance parameters $\boldsymbol{\theta} = (\sigma^2 = 1, \ell = 0.1)$.



(a) $\sigma^2 = 0.1$, $\ell = 0.1$ (b) $\sigma^2 = 10$, $\ell = 0.1$ (c) $\sigma^2 = 1$, $\ell = 0.5$

Figure 2.2: Effect of different covariance parameters $\boldsymbol{\theta} = (\sigma^2, \ell)$ on unconditional GP samples with SE kernel functions.

functions. Contrarily to the class C^∞, the class C^κ corresponds to the space of κ-times differentiable functions and the class C corresponds to the space of continuous functions. Notice that the kernels from Table 2.1 depend only on two covariance parameters $\boldsymbol{\theta} = (\sigma^2, \ell)$ with variance parameter σ^2 and length-scale parameter ℓ. For instance, σ^2 can be viewed as a scale parameter for the output; and ℓ as a scale parameter for the input. The effect of different kernels and covariance parameters on unconditional GP samples can be seen in Figures 2.1 and 2.2, respectively.

For the multidimensional case (i.e. $d \geq 2$), valid kernel functions can be obtained by the tensor product of one-dimensional ones:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \prod_{i=1}^{d} k_i(x_i, {x_i}'),$$

where $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{D}$, and k_i corresponds to the kernel associated to the i-th dimension. Note that the expression above is valid for any class of kernel k_i. We refer to (Genton, 2001; Rasmussen and Williams, 2005) for a further discussion about the construction of valid kernel functions (e.g. locally stationary kernels, non-stationary kernels).

2.2.2 Gaussian process regression

Now, we focus on conditioning a GP Y to a finite number of observations (conditioning points). Consider the notation in subsection 2.2.1. Let $\mathbf{Y}_n = \left[Y(\boldsymbol{x}_1), \cdots, Y(\boldsymbol{x}_n)\right]^\top$ and $\mathbf{Y}_* = \left[Y(\boldsymbol{x}_1^*), \cdots, Y(\boldsymbol{x}_m^*)\right]^\top$ be Gaussian vectors extracted from Y. Then, the joint vector $(\mathbf{Y}_n, \mathbf{Y}_*)$ is also Gaussian:

$$\begin{bmatrix}\mathbf{Y}_n \\ \mathbf{Y}_*\end{bmatrix} = \mathcal{N}\left(\begin{bmatrix}\mathbf{0} \\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n} & \mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_*} \\ \mathbf{K}_{\mathbf{Y}_*,\mathbf{Y}_n} & \mathbf{K}_{\mathbf{Y}_*,\mathbf{Y}_*}\end{bmatrix}\right), \tag{2.3}$$

with covariance matrices $\mathbf{K}_{\cdot,\cdot} = \operatorname{cov}\{\cdot,\cdot\}$ satisfying $\mathbf{K}_{\mathbf{Y}_*,\mathbf{Y}_n} = \mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_*}^\top$. Finally, according to the properties on GPs (see, e.g., Rasmussen and Williams, 2005, appendix A.2), the conditional distribution of the Gaussian vector $\mathbf{Y}_*$, knowing $\mathbf{Y}_n = \boldsymbol{y}$ for a given observation vector $\boldsymbol{y} = [y_1, \cdots, y_n]^\top$, is given by

$$\mathbf{Y}_* | \{\mathbf{Y}_n = \boldsymbol{y}\} \sim \mathcal{N}(\boldsymbol{\mu},\ \boldsymbol{\Sigma}). \tag{2.4}$$

with conditional parameters,

$$\boldsymbol{\mu} = \mathbf{K}_{\mathbf{Y}_*,\mathbf{Y}_n}\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n}^{-1}\ \boldsymbol{y}, \quad \boldsymbol{\Sigma} = \mathbf{K}_{\mathbf{Y}_*,\mathbf{Y}_*} - \mathbf{K}_{\mathbf{Y}_*,\mathbf{Y}_n}\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n}^{-1}\mathbf{K}_{\mathbf{Y}_*,\mathbf{Y}_n}^\top. \tag{2.5}$$

Furthermore, one can compute the $q_{\%}$ conditional confidence intervals (Hyndman and Fan, 1996):

$$\mathrm{CI}_{q_{\%}} = \left[\boldsymbol{\mu} - z_{q_{\%}}\boldsymbol{\sigma}, \boldsymbol{\mu} + z_{q_{\%}}\boldsymbol{\sigma}\right], \tag{2.6}$$

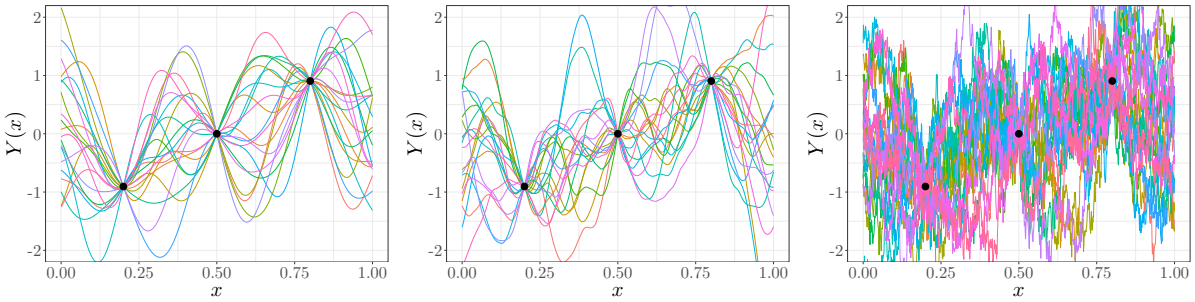
with the vector of standard deviations $\boldsymbol{\sigma} = \sqrt{\operatorname{diag}(\boldsymbol{\Sigma})}$, and $z_{q_{\%}}$ the value from the standard normal distribution for the selected confidence level. For 90%, 95% and 99% conditional confidence intervals, we have the z-values $z_{90\%} = 1.645$, $z_{95\%} = 1.96$ and $z_{99\%} = 2.576$.

The effect of adding conditioning points into the GPs in Figure 2.1 can be seen in Figure 2.3. Figure 2.4 shows the conditional mean and conditional variances obtained by the resulting GP with a SE kernel.

Finally, noise effects can also be added in GP frameworks. Consider the Gaussian vector $\mathbf{Y}_*$, conditioned to $\mathbf{Y}_n + \boldsymbol{\varepsilon} = \boldsymbol{y}$ for a given observation vector $\boldsymbol{y}$ and an additive Gaussian white noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$ with noise variance τ^2. Note that $\varepsilon_1, \cdots, \varepsilon_n$ were assumed to be independent, and independent of Y. Then, the conditional distribution $\mathbf{Y}_* | \{\mathbf{Y}_n + \boldsymbol{\varepsilon} = \boldsymbol{y}\}$ is as in (2.4), where the covariance matrix $\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n}$ in (2.5) is replaced by $\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n} + \tau^2\mathbf{I}$.

2.2.3 Covariance parameter estimation

The covariance functions of GP models are usually unknown and have to be estimated. Let $\{k_{\boldsymbol{\theta}}; \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, with $\boldsymbol{\Theta} \subset \mathbb{R}^p$, be a parametric family of covariance functions. Here, we assume that Y is a centred GP with covariance function $k_{\boldsymbol{\theta}^*}$ for an unknown $\boldsymbol{\theta}^* \in \boldsymbol{\Theta}$, and we consider the problem of estimating $\boldsymbol{\theta}^*$. We denote $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$ as covariance parameters.



(a) SE (b) Matérn 5/2 (c) Exponential

Figure 2.3: Samples of conditional GPs using various types of kernels. Conditioning observations at $x_1 = 0.2$, $x_2 = 0.5$, and $x_3 = 0.8$ are shown in black dots. The covariance parameters are fixed to $\boldsymbol{\theta} = (\sigma^2 = 1, \ell = 0.1)$.

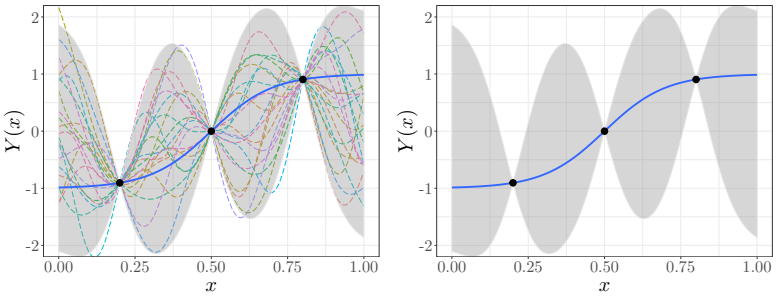


Figure 2.4: Conditional GP of Figure 2.3(a). The panel shows: the conditional mean function (blue solid line), two standard deviation conditional intervals (grey region), and conditional GP realisations (dashed lines).

Commonly, the parameters $\boldsymbol{\theta}^*$ are estimated by maximising the Gaussian likelihood $P_{\boldsymbol{\theta}}(\mathbf{Y}_n)$ with respect to (w.r.t.) $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. Let $\mathcal{L}_n(\boldsymbol{\theta})$ be the log-likelihood of $\boldsymbol{\theta}$:

$$\mathcal{L}_n(\boldsymbol{\theta}) = \log P_{\boldsymbol{\theta}}(\mathbf{Y}_n) = -\frac{1}{2}\log(\det(\mathbf{K}_{\boldsymbol{\theta}})) - \frac{1}{2}\mathbf{Y}_n^\top \mathbf{K}_{\boldsymbol{\theta}}^{-1}\mathbf{Y}_n - \frac{n}{2}\log 2\pi, \tag{2.7}$$

with $\mathbf{K}_{\boldsymbol{\theta}} = (k_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{x}_j))_{1\leq i,j\leq n}$. Then, the maximum likelihood estimator (MLE) is given by

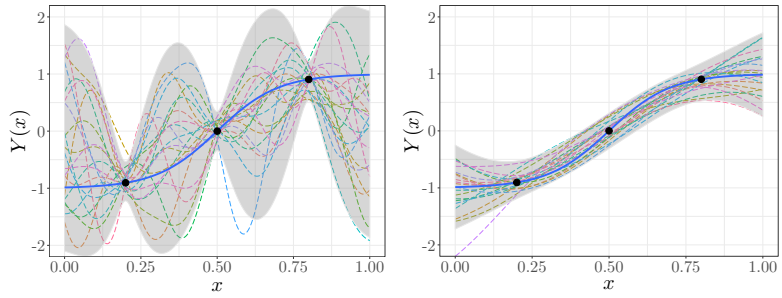
$$\widehat{\boldsymbol{\theta}}_{\text{MLE}} \in \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \ \mathcal{L}_n(\boldsymbol{\theta}). \tag{2.8}$$

By maximising (2.7), we are looking for a set of parameters $\boldsymbol{\theta}$ that improves the ability of the model to explain the data. Figure 2.5 shows the resulting conditional GPs when the covariance parameters are: (a) manually fixed to $\boldsymbol{\theta} = (\sigma^2 = 1, \ell = 0.1)$, or (b) estimated via maximum likelihood (ML).

Note that other approaches based on cross-validation can also be used for the covariance parameter estimation of GPs (see, e.g. Bachoc, 2013; Rasmussen and Williams, 2005; Stein, 1999), but we focus on ML estimators in the later chapters (see Chapter 7).

2.2.4 Curse of dimensionality

The main drawback of standard GP implementations is related to the curse of dimensionality: it typically scales in $\mathcal{O}(n^2)$ in space and $\mathcal{O}(n^3)$ in time with n the number



(a) $\boldsymbol{\theta} = (\sigma^2 = 1, \ell = 0.1)$ (b) $\widehat{\boldsymbol{\theta}}_{\text{MLE}} = (\widehat{\sigma}^2 = 0.90, \widehat{\ell} = 0.42)$

Figure 2.5: Conditional GP from Figure 2.4. Figure 2.5(b) shows the conditional process when the covariance parameters $\boldsymbol{\theta}$ are estimated via ML. On both models, an additive Gaussian white noise was added with a fixed noise variance $\tau^2 = 0.03$.

of observations (Rasmussen and Williams, 2005; Titsias, 2009). The computational cost results mainly from the inversion of the covariance matrix $\mathbf{K}_{\boldsymbol{\theta}}$ when training GP models using (2.8). More sophisticated GP developments have been proposed to deal with the intractability when n is large. First, one may consider further assumptions on the process looking for computational simplifications. An example can be seen when considering additive processes (Durrande et al., 2012; Duvenaud et al., 2011; Van der Wilk et al., 2017). In that case, the computational gain is in the fact that an exponentially increasing number of design points $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathbb{R}^d$ is not required to cover the input space when d increases. Second, most common GP frameworks are based on sparse representations using inducing points (Hensman et al., 2017, 2013; Quiñonero Candela and Rasmussen, 2005; Snelson and Ghahramani, 2006; Titsias, 2009). There, approximations based on a small set of m inducing variables are constructed allowing to reduce the time complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2)$ for $m << n$. The main difference between those sparse approaches relies in the strategy used to choose the inducing variables. Finally, specific covariance structures leading to more efficient computations can be exploited: e.g., in state space models (Särkkä, 2013), Gaussian Markov models (Durrande et al., 2019), etc. In Chapter 6, we focus on the class of models accounting for both additive structures and inequality constraints.

2.3 Additive models

Nowadays, many statistical regression models are based on additive structures of the form:

$$f(x_1, \cdots, x_d) = f_1(x_1) + \cdots + f_d(x_d). \tag{2.9}$$

Although having such structure may lead to more "rigid" models compared to non-additive ones, it results in simple frameworks that can be easily scaled in high dimensions (Buja et al., 1989; Hastie and Tibshirani, 1986). Generalised additive models (GAMs) (Hastie and Tibshirani, 1986), and additive GPs (Durrande et al., 2012; Duvenaud et al., 2011), are the most common models successfully applied in a wide range of applications. Furthermore, the latter can also be seen as a generalisation of GAMs that allow to quantify uncertainties. As shown by Duvenaud et al. (2011) and Durrande et al. (2012),

additive GPs can significantly improve modelling efficiency and has major advantages for interpretability. As our general framework is about GPs, we focus on additive GP models.

2.3.1 Additive Gaussian processes

First, we consider the additive GP $\{Y(\boldsymbol{x}); \boldsymbol{x} \in \mathcal{D}\}$ given by

$$Y(\boldsymbol{x}) = Y_1(x_1) + \cdots + Y_d(x_d), \tag{2.10}$$

where $x_1, \cdots, x_d \in [0,1]$. We assume that $Y_1, \cdots, Y_d$ are independent GPs with mean zero and covariance functions $k_1, \cdots, k_d$, respectively. Then, cross-covariance terms are equal to zero, i.e. $\operatorname{cov}\{Y_p(x_p), Y_q(x_q)\} = 0$ for $p, q = 1, \cdots, d$, and $p \neq q$. Thus, the covariance function k of Y is given by

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{p=1}^{d} k_p(x_p, x_p'), \tag{2.11}$$

with $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{D}$. Notice that the extension of (2.10) to dependent sub-processes Y_p is straightforward. In that case, cross-covariance terms $k_{p,q}(\cdot,\cdot) = \operatorname{cov}\{Y_p(\cdot), Y_q(\cdot)\} \neq 0$ will appear in (2.11). Furthermore, developments considering block-additivity can also be considered leading to minor changes. For example, consider a block-additive process $Y(x_1, x_2, x_3) = Y_1(x_1) + Y_{2,3}(x_2, x_3)$ with independent GPs Y_1 and $Y_{2,3}$ on $\mathbb{R}$ and $\mathbb{R}^2$, respectively. Denote k_1 and $k_{2,3}$ the covariance functions of $Y_1, Y_{2,3}$, respectively. Then, (2.11) is given by the sum of kernels $k(\boldsymbol{x}, \boldsymbol{x}') = k_1(x_1, x_1') + k_{2,3}(x_2, x_3; x_2', x_3')$. We refer to (Durrande et al., 2012; Duvenaud et al., 2011; Fruth, 2015; Muehlenstaedt et al., 2012; Van der Wilk et al., 2017) for further details on the selection of groups for additive GPs per blocks.

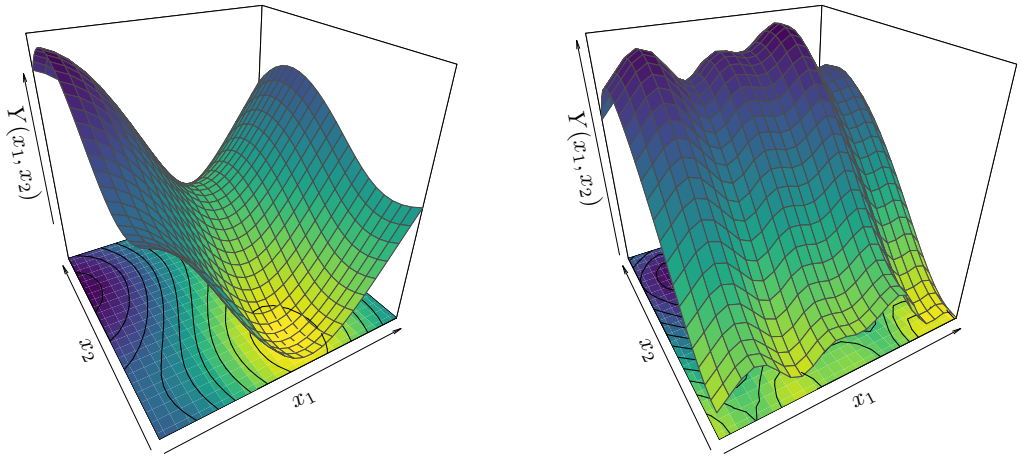
As in (Durrande et al., 2012), here we consider covariance functions of the form $k_p(x_p, x_p') = \sigma_p^2 c_{p,\ell_p}(x_p, x_p')$ for $p = 1, \cdots, d$, with c_{p,ℓ_p} the p-th correlation function with length-scale ℓ_p. Thus, (2.11) can be written as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{p=1}^{d} \sigma_p^2 c_{p,\ell_p}(x_p, x_p'), \tag{2.12}$$

Note that, unlike standard GP models, the additivity allows us to consider different values of variance parameters along each input coordinate. This implies that, for d dimensional input spaces, $2 \times d$ covariance parameters (d variances and d length-scales parameters) have to be estimated rather than the required $d+1$ parameters for non-additive models (a single variance term and d length-scales). Figure 2.6 shows 2D simulations from the additive GP $Y(x_1, x_2) = Y_1(x_1) + Y_2(x_2)$. We used an additive SE kernel,

$$k(x_1, x_2; x_1', x_2') = \sigma_1^2 \exp\left\{-\frac{(x_1 - x_1')^2}{2\ell_1^2}\right\} + \sigma_2^2 \exp\left\{-\frac{(x_2 - x_2')^2}{2\ell_2^2}\right\},$$

with covariance parameters $\boldsymbol{\theta} = ((\sigma_1^2, \ell_1), (\sigma_2^2, \ell_2))$.



(a) $(\sigma_1^2, \ell_1) = (\sigma_2^2, \ell_2) = (1, 0.3)$ (b) $(\sigma_1^2, \ell_1) = (0.1, 0.1)$, $(\sigma_2^2, \ell_2) = (1, 0.3)$

Figure 2.6: Examples of 2D trajectories from additive GPs $Y(x_1, x_2) = Y_1(x_1) + Y_2(x_2)$ with covariance parameters $\boldsymbol{\theta} = ((\sigma_1^2, \ell_1), (\sigma_2^2, \ell_2))$.

2.3.2 Conditioning to interpolation constraints

Consider the model in (2.10). Here, we aim at satisfying the interpolations conditions:

$$Y(\boldsymbol{x}_i) = y_i,$$

with $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathcal{D}$ and observations $y_i \in \mathbb{R}$ for $i = 1, \cdots, n$. As in subsection 2.2.2, let $\mathbf{Y}_n = \left[Y(\boldsymbol{x}_1), \cdots, Y(\boldsymbol{x}_n)\right]^\top$ and $\mathbf{Y}_* = \left[Y(\boldsymbol{x}_1^*), \cdots, Y(\boldsymbol{x}_m^*)\right]^\top$ be Gaussian vectors extracted from Y. Then, according to Section 2.2, and using (2.11), the distribution of $\mathbf{Y}_* | \{\mathbf{Y}_n = \boldsymbol{y}\}$ is also Gaussian with conditional parameters given by,

$$\begin{aligned} \boldsymbol{\mu} &= \left(\sum_{p=1}^{d} \mathbf{K}^{(p)}_{\mathbf{Y}_*, \mathbf{Y}_n}\right)\left(\sum_{p=1}^{d} \mathbf{K}^{(p)}_{\mathbf{Y}_n, \mathbf{Y}_n}\right)^{-1} \boldsymbol{y}, \\ \boldsymbol{\Sigma} &= \left(\sum_{p=1}^{d} \mathbf{K}^{(p)}_{\mathbf{Y}_*, \mathbf{Y}_*}\right) - \left(\sum_{p=1}^{d} \mathbf{K}^{(p)}_{\mathbf{Y}_*, \mathbf{Y}_n}\right)\left(\sum_{p=1}^{d} \mathbf{K}^{(p)}_{\mathbf{Y}_n, \mathbf{Y}_n}\right)^{-1}\left(\sum_{p=1}^{d} \mathbf{K}^{(p)}_{\mathbf{Y}_*, \mathbf{Y}_n}\right)^{\top}, \end{aligned} \tag{2.13}$$

with covariance matrices $\mathbf{K}^{(p)}_{\mathbf{Y}_n, \mathbf{Y}_n} = (k_p(\boldsymbol{x}_i, \boldsymbol{x}_j))_{1 \leq i,j \leq n}$, $\mathbf{K}^{(p)}_{\mathbf{Y}_n, \mathbf{Y}_*} = (k_p(\boldsymbol{x}_i, \boldsymbol{x}_j^*))_{1 \leq i \leq n, 1 \leq j \leq m}$ and $\mathbf{K}^{(p)}_{\mathbf{Y}_*, \mathbf{Y}_*} = (k_p(\boldsymbol{x}_i^*, \boldsymbol{x}_j^*))_{1 \leq i,j \leq m}$ for $p = 1, \cdots, d$. Note that the expressions in (2.13) are exactly equal to the ones in (2.5) with $\mathbf{K}_{\mathbf{Y}_n, \mathbf{Y}_n} = \sum_{p=1}^{d} \mathbf{K}^{(p)}_{\mathbf{Y}_n, \mathbf{Y}_n}$, $\mathbf{K}_{\mathbf{Y}_*, \mathbf{Y}_n} = \sum_{p=1}^{d} \mathbf{K}^{(p)}_{\mathbf{Y}_*, \mathbf{Y}_n}$ and $\mathbf{K}_{\mathbf{Y}_*, \mathbf{Y}_*} = \sum_{p=1}^{d} \mathbf{K}^{(p)}_{\mathbf{Y}_*, \mathbf{Y}_*}$. One must note that the matrix $\mathbf{K}_{\mathbf{Y}_n, \mathbf{Y}_n}$ can be non-invertible for some choices of $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$. However, a solution to this drawback is to consider noisy observations (see subsection 2.3.3 for more details).

From (2.13), we have that the mean predictor $\boldsymbol{\mu}$ can also be expressed as the sum of

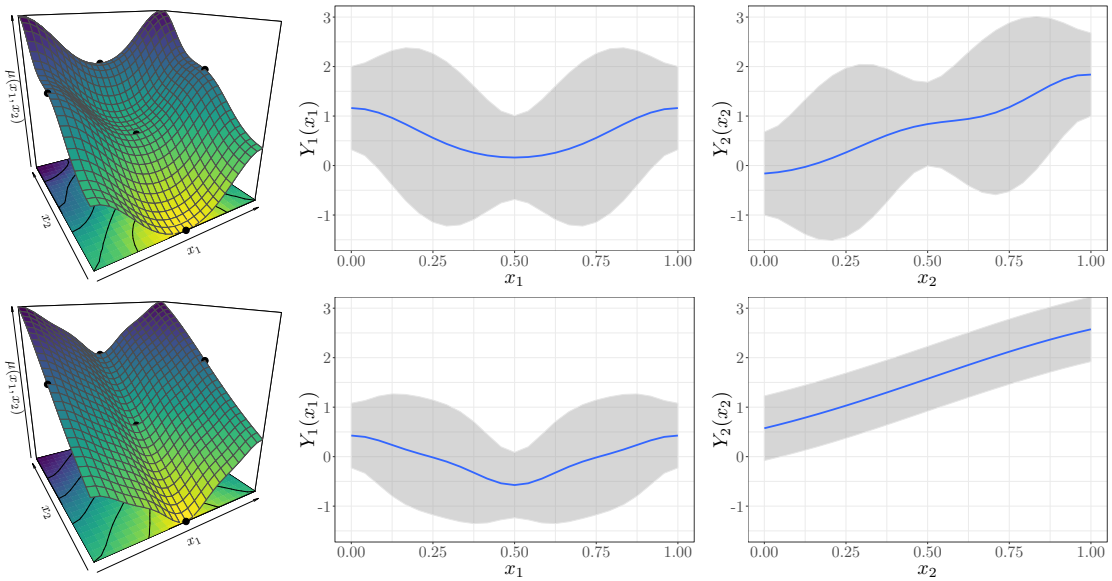


Figure 2.7: 2D conditional additive GPs. Results are shown either (top) with fixed covariance parameters $\boldsymbol{\theta} = ((\sigma_1^2 = 1, \ell_1 = 0.2), (\sigma_2^2 = 1, \ell_2 = 0.2))$, or (bottom) with the estimated ones via ML: $\widehat{\boldsymbol{\theta}}_{\text{MLE}} = ((\widehat{\sigma}_1^2 = 0.34, \widehat{\ell}_1 = 0.12), (\widehat{\sigma}_2^2 = 3.04, \widehat{\ell}_2 = 1))$. Each row shows: (left) the conditional mean function $\mu(x_1, x_2) = \mu_1(x_1) + \mu_2(x_2)$, and the sub-models (centre) μ_1 and (right) μ_2. For the conditional mean, black dots represent the interpolation points. For the sub-models, the conditional mean functions μ_1, μ_2 (blue solid lines) and the 90% conditional interval (grey region) are shown.

mean terms:

$$\boldsymbol{\mu} = \mathbf{K}_{\mathbf{Y}_*,\mathbf{Y}_n}\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n}^{-1}\, \boldsymbol{y} = \mathbf{K}^{(1)}_{\mathbf{Y}_*,\mathbf{Y}_n}\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n}^{-1}\, \boldsymbol{y} + \cdots + \mathbf{K}^{(d)}_{\mathbf{Y}_*,\mathbf{Y}_n}\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n}^{-1}\, \boldsymbol{y} = \boldsymbol{\mu}_1 + \cdots + \boldsymbol{\mu}_d,$$

with sub-models $\boldsymbol{\mu}_p = \mathbf{K}^{(p)}_{\mathbf{Y}_*,\mathbf{Y}_n}\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n}^{-1}\, \boldsymbol{y}$ for $p = 1, \cdots, d$ (Durrande et al., 2012). On the other hand, the conditional covariance matrix $\boldsymbol{\Sigma}$ is not the sum of covariance matrices of individual models. But we have:

$$\boldsymbol{\Sigma} = \mathbf{K}_{\mathbf{Y}_*,\mathbf{Y}_*} - \mathbf{K}_{\mathbf{Y}_*,\mathbf{Y}_n}\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n}^{-1}\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_*} = \sum_{p=1}^{d} \boldsymbol{\Sigma}_p,$$

where $\boldsymbol{\Sigma}_p = \mathbf{K}^{(p)}_{\mathbf{Y}_*,\mathbf{Y}_*} - \mathbf{K}^{(p)}_{\mathbf{Y}_*,\mathbf{Y}_n}(\sum_{\nu=1}^{d}\mathbf{K}^{(\nu)}_{\mathbf{Y}_n,\mathbf{Y}_n})^{-1}(\mathbf{K}^{(p)}_{\mathbf{Y}_*,\mathbf{Y}_n})^\top$ is the conditional covariance matrix of the p-th sub-model. Note that the novelty here lies in the term $(\sum_{p=1}^{d}\mathbf{K}^{(p)}_{\mathbf{Y}_n,\mathbf{Y}_n})^{-1}$ instead of $(\mathbf{K}^{(p)}_{\mathbf{Y}_n,\mathbf{Y}_n})^{-1}$. The two above expressions mean that $Y_1, \cdots, Y_d$ are independent conditionally to the observation data.

Figure 2.7 shows a 2D example of a conditional additive GP model. As interpolation conditions, we evaluate the function $(x_1, x_2) \mapsto 4(x_1 - 0.5)^2 + 2x_2$ at points $(x_i, y_i)_{1\leq i\leq 5}$: $(0.5, 0)$, $(0.5, 0.5)$, $(0.5, 1)$, $(0, 0.5)$, and $(1, 0.5)$. We fix a noise variance $\tau^2 = 10^{-5}$ and use an additive SE kernel. Results are shown either with fixed covariance parameters $\boldsymbol{\theta} = ((\sigma_1^2, \ell_1), (\sigma_2^2, \ell_2))$, or with the estimated ones via ML.

2.3.3 Remarks

- The parameters $\boldsymbol{\theta} = ((\sigma_1^2, \ell_1), \cdots, (\sigma_d^2, \ell_d))$ of additive GPs can be estimated via ML as in (2.7) with covariance $\mathbf{K}_{\boldsymbol{\theta}} = (k(\boldsymbol{x}^i, \boldsymbol{x}^j))_{1 \leq i,j \leq n}$, for $\boldsymbol{x}^i, \boldsymbol{x}^j \in \mathcal{D}$, and kernel k as in (2.12). In that case, the gradients of $\mathbf{K}_{\boldsymbol{\theta}}$ w.r.t. $\boldsymbol{\theta}$ are equal to:

$$\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} = \frac{\partial \mathbf{K}_{\boldsymbol{\theta}}^{(1)}}{\partial \boldsymbol{\theta}} + \cdots + \frac{\partial \mathbf{K}_{\boldsymbol{\theta}}^{(d)}}{\partial \boldsymbol{\theta}},$$

  with $\mathbf{K}_{\boldsymbol{\theta}}^{(p)} = (k_p(x_p^i, x_p^j))_{1 \leq i,j \leq n}$ for $p = 1, \cdots, d$. Since $\mathbf{K}_{\boldsymbol{\theta}}^{(p)}$ depends only on $\boldsymbol{\theta}_p = (\sigma_p^2, \ell_p)$, then we only need to compute partial derivatives of the form $\frac{\partial}{\partial \boldsymbol{\theta}_p} \mathbf{K}_{\boldsymbol{\theta}}^{(p)}$ and stack them in a vector of gradients. Then, $\widehat{\boldsymbol{\theta}}_{\text{MLE}}$ can be estimated via a gradient-based optimisation.

- Noise effects can also be considered. Consider the Gaussian vector $\mathbf{Y}_*$, conditional on $\mathbf{Y}_n + \boldsymbol{\varepsilon} = \boldsymbol{y}$ for a given observation vector $\boldsymbol{y}$ and an additive Gaussian white noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \tau^2 \mathbf{I})$ with noise variance τ^2. Assume $\varepsilon_1, \cdots, \varepsilon_n$ to be independent, and independent of Y. Thus, the conditional parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ follow (2.13) but replacing $\sum_{p=1}^d \mathbf{K}_{\mathbf{Y}_n, \mathbf{Y}_n}^{(p)}$ by $\sum_{p=1}^d \mathbf{K}_{\mathbf{Y}_n, \mathbf{Y}_n}^{(p)} + \tau^2 \mathbf{I}$.

2.4 Conclusions

In this chapter, we introduced the notations and definitions used along the manuscript. We started from the classical point of view, defining key terms of standard GP regression models. We then briefly explained how to adapt GP models for additive functions. As shown in different illustrations, the versatility of GPs relies mainly on the choice of the covariance function used for the prior distribution. Different assumptions can be considered depending on previous experiences or beliefs about the types of functions we expect to observe (e.g. smoothness, stationarity).

The main limitation of GP models relies in the intractability for large datasets because the time complexity scales as $\mathcal{O}(n^3)$ and the storage as $\mathcal{O}(n^2)$. This drawback is more sensitive in high dimensions where an exponentially increasing number of observations is required to cover the input space and for the covariance parameter estimation (Durrande et al., 2012; Rasmussen and Williams, 2005). Recent GP developments have been proposed in order to mitigate this limitation, allowing implementations in high dimensions and thousands of observations (e.g. additive GPs, Durrande et al. (2012); Van der Wilk et al. (2017); sparse GPs, Hensman et al. (2013); Titsias (2009)).

Part III

Gaussian Processes under Inequality Constraints

Chapter 3

Gaussian Process Regression under Linear Inequality Constraints

Contents

3.1 Introduction

Despite the reliable performance of GPs, they provide less realistic uncertainties when physical systems satisfy inequality constraints (Da Veiga and Marrel, 2012; Maatouk and Bay, 2017; Riihimäki and Vehtari, 2010). Quantifying properly those uncertainties is crucial for understanding real-world phenomena. For example, in nuclear safety criticality assessment, experimental settings typically demand expensive and risky procedures to

evaluate neutron productions. Hence, emulators are required to infer these production rates and should assume a priori that the output is positive (and usually monotonic) w.r.t. a given set of input parameters (see Section 3.4). Therefore, both conditions have to be considered in uncertainty quantification in order to obtain more accurate predictions.

In this chapter, the main contributions are twofold. First, we further investigate the finite-dimensional approximation of GPs in (Maatouk and Bay, 2017) to account for general linear inequality constraints. This leads to versatile models that can be used for a broad range of applications. Furthermore, in contrast to (Maatouk and Bay, 2017), our framework does not require to compute kernel derivatives, which implies there is no restrictions on the choice of the kernel function (e.g. as for monotonicity or convexity). Second, we show that using the Hamiltonian Monte Carlo (HMC) sampler suggested in subsection 3.3.2 results in high effective sample rates with reasonable running times. On numerical experiments, the proposed framework together with the HMC sampler provides accurate and efficient results in the order of seconds (compared to tens of minutes using Maatouk and Bay, 2017), leading to models that can be easily used in practice.

The ability of the model to account for multiple constraints (e.g. boundedness and monotonicity) or more complex ones (see, e.g., Figure 3.4) is tested on various examples. We also assess its performance on both data fitting and uncertainty quantification on a 2D nuclear application where both boundedness and monotonicity are taken into account.

This chapter is based on the journal paper:

- López-Lopera, A. F., Bachoc, F., Durrande, N., and Roustant, O. (2018). Finite-dimensional Gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1224–1255.

3.2 Finite-dimensional approximation of Gaussian processes

We first briefly describe the framework proposed by Maatouk and Bay (2017) that account for inequality conditions (either boundedness, monotonicity or convexity). We start from the finite-dimensional approximation of GPs in 1D, summarising some key properties. Then, we finish with the extension of the model to higher dimensions by tensorisation.

3.2.1 Finite-dimensional approximation in 1D

Let $\{Y(x); x \in \mathcal{D}\}$ be a centred GP with covariance function k. Consider a compact input space $\mathcal{D} = [0, 1]$, and a set of knots $t_1, \cdots, t_m \in \mathbb{R}$. For simplicity, we consider equispaced knots $t_j = (j-1)\Delta_m$ with $\Delta_m = 1/(m-1)$, but this assumption can be relaxed (see, e.g., Chapter 5). Then, define a finite-dimensional GP, denoted by Y_m, as the piecewise linear interpolation of Y at knots $t_1, \cdots, t_m$:

$$Y_m(x) = \sum_{j=1}^{m} Y(t_j)\phi_j(x), \tag{3.1}$$

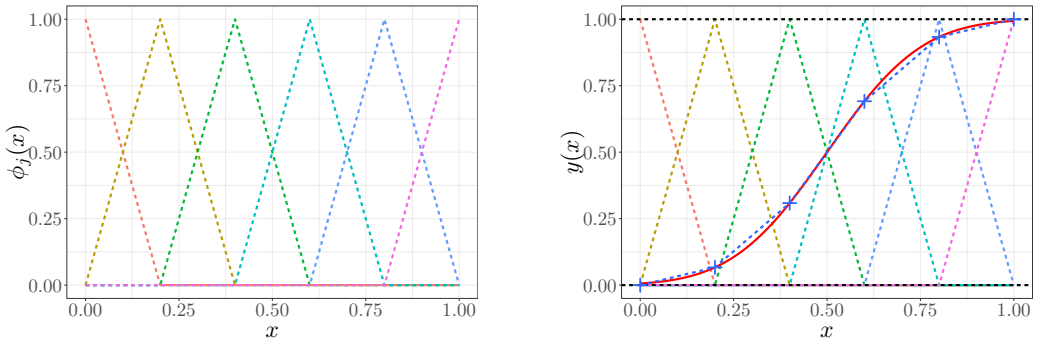


Figure 3.1: Illustration of the finite-dimensional approximation in (3.1). (Left) Hat functions ϕ_j for $j = 1, \cdots, 6$. (Right) Approximation of the function $y(x) = \Phi(\frac{x-0.5}{0.2})$, where Φ is the standard normal cumulative distribution function. Red solid and blue dashed lines are the function y, and its finite approximation with six knots given by blue crosses, respectively. Horizontal black dashed lines denote the bounds.

where $\phi_1, \cdots, \phi_m$ are hat basis functions given by

$$\phi_j(x) := \begin{cases} 1 - \left|\frac{x-t_j}{\Delta_m}\right| & \text{if } \left|\frac{x-t_j}{\Delta_m}\right| \leq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

Note that quality of resolution of the approximation in (3.1) depends on the number of knots m. For larger values of m, better is the resolution of Y_m. Figure 3.1 illustrates the finite-dimensional approach in (3.1) for a deterministic function that satisfies both boundedness and monotonicity constraints.

We aim at computing the distribution of Y_m, conditionally on $Y_m \in \mathcal{E}$, where $\mathcal{E}$ is a convex set of functions defined by some inequality constraints. For instance, according to (Maatouk and Bay, 2017), we may have

$$\mathcal{E} = \mathcal{E}_\kappa := \begin{cases} \{f \in C(\mathcal{D}, \mathbb{R}) \text{ s.t. } l \leq f(x) \leq u, \ \forall x \in \mathcal{D}\} & \text{if } \kappa = 0, \\ \{f \in C(\mathcal{D}, \mathbb{R}) \text{ s.t. } f \text{ is non-decreasing}\} & \text{if } \kappa = 1, \\ \{f \in C(\mathcal{D}, \mathbb{R}) \text{ s.t. } f \text{ is convex}\} & \text{if } \kappa = 2, \end{cases} \tag{3.3}$$

which corresponds to boundedness, monotonicity, and convexity constraints, respectively.

Now, let $\xi_j := Y(t_j)$ for $j = 1, \cdots, m$. As discussed in (Maatouk, 2015), the main benefit of using the hat functions in (3.2) and the finite-dimensional approximation Y_m, is that satisfying the inequality conditions $Y_m \in \mathcal{E}$, is equivalent to satisfying only a finite number of inequality constraints.

Proposition 3.1 (Maatouk (2015)) *Define a finite-dimensional GP, denoted by Y_m, as the piecewise linear interpolation of a centred GP Y at knots $t_1, \cdots, t_m \in \mathbb{R}$. For many natural choices of $\mathcal{E}$ (such as $\mathcal{E} = \mathcal{E}_\kappa$, for $\kappa = 0, 1, 2$), we have*

$$Y_m \in \mathcal{E} \quad \Leftrightarrow \quad \boldsymbol{\xi} \in \mathcal{C}, \tag{3.4}$$

where $\mathcal{C}$ is a convex set of $\mathbb{R}^m$, and $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_m]^\top$.

For instance, for the convex set $\mathcal{E}_\kappa$ in (3.3), we have

$$\mathcal{C} = \mathcal{C}_\kappa := \begin{cases} \{\mathbf{c} \in \mathbb{R}^m;\ \forall\ j = 1, \cdots, m\ :\ l \leq c_j \leq u\} & \text{if } \kappa = 0, \\ \{\mathbf{c} \in \mathbb{R}^m;\ \forall\ j = 2, \cdots, m\ :\ c_j \geq c_{j-1}\} & \text{if } \kappa = 1, \\ \{\mathbf{c} \in \mathbb{R}^m;\ \forall\ j = 3, \cdots, m\ :\ c_j - c_{j-1} \geq c_{j-1} - c_{j-2}\} & \text{if } \kappa = 2. \end{cases} \tag{3.5}$$

3.2.2 Conditioning to interpolation and inequality constraints

Consider the finite-dimensional representation of GPs as in (3.1), given the interpolation and inequality constraints:

$$Y_m(x) = \sum_{j=1}^{m} \xi_j \phi_j(x), \quad \text{s.t.} \quad \begin{cases} Y_m(x_i) = y_i & \text{(interpolation conditions)}, \\ Y_m \in \mathcal{E} & \text{(inequality conditions)}, \end{cases} \tag{3.6}$$

where $x_i \in \mathcal{D}$ and $y_i \in \mathbb{R}$ for $i = 1, \cdots, n$. Note from (3.6) that noise-free observations are considered but a noise effect can be included assuming $Y_m(x_i) + \varepsilon_i = y_i$ with Gaussian noise $\varepsilon_i \sim \mathcal{N}(0, \tau^2)$. We refer to Chapter 4 for further details on how to incorporate noise effects in constrained GP models.

Given a design of experiment (DoE) $\boldsymbol{x} = [x_1, \cdots, x_n]^\top$, we have matricially:

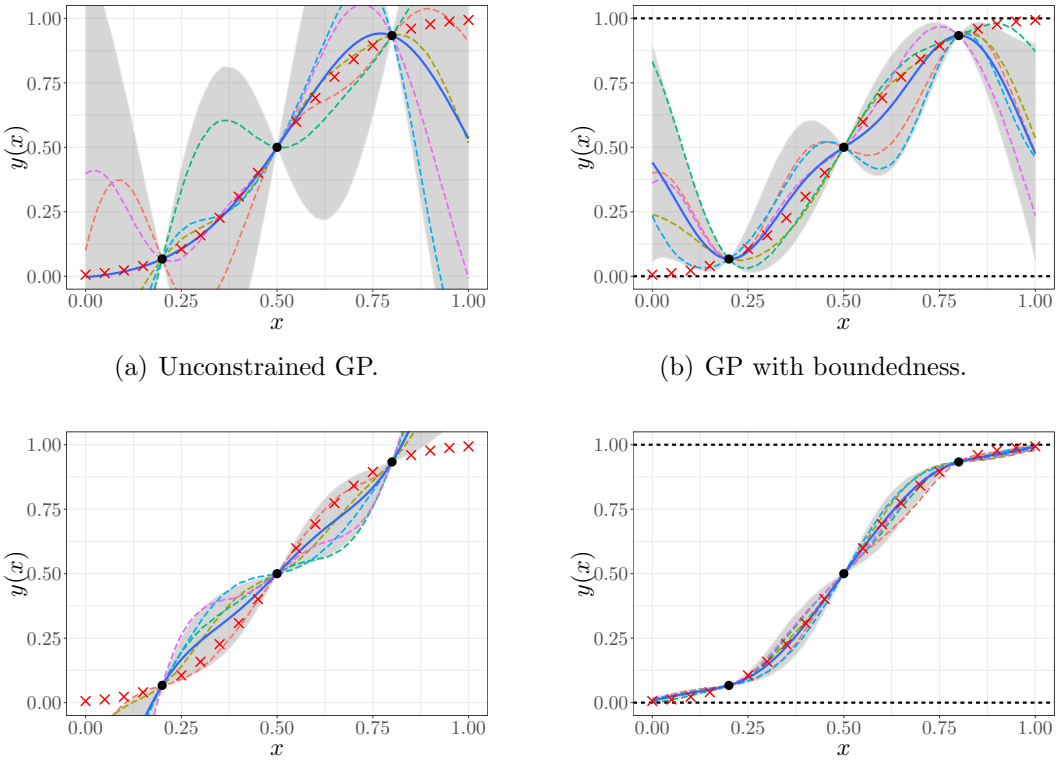
$$\mathbf{Y}_m = \big[Y_m(x_1),\ \ \cdots,\ \ Y_m(x_n)\big]^\top = \boldsymbol{\Phi}\boldsymbol{\xi},$$

where $\boldsymbol{\Phi}$ is the $n \times m$ matrix defined by $\boldsymbol{\Phi}_{i,j} = \phi_j(x_i)$. From (3.4), the conditional distribution of Y_m, given $Y_m \in \mathcal{E}$ and $Y_m(x_i) = y_i$ for $i = 1, \cdots, n$, can be obtained from the conditional distribution of $\boldsymbol{\xi}$ given $\boldsymbol{\xi} \in \mathcal{C}$ and $\boldsymbol{\Phi}\boldsymbol{\xi} = \boldsymbol{y}$.

Observe that the vector $\boldsymbol{\xi}$ of the values at the knots is a centred Gaussian vector with covariance matrix $\boldsymbol{\Gamma} = (k(t_i, t_j))_{1 \leq i,j \leq m}$. Then, the distribution of $\boldsymbol{\xi}$ given both interpolation and inequality conditions is truncated multinormal:

$$\boldsymbol{\xi} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Gamma}\right) \quad \text{s.t.} \quad \begin{cases} \boldsymbol{\Phi}\boldsymbol{\xi} = \boldsymbol{y} & \text{(interpolation conditions)}, \\ \boldsymbol{\xi} \in \mathcal{C} & \text{(inequality conditions)}. \end{cases} \tag{3.7}$$

Figure 3.2 shows different Gaussian models for the example of Figure 3.1. We use a SE kernel with parameters ($\sigma^2 = 1,\ \ell = 0.2$) and fix $m = 100$. The posterior distribution is approximated via HMC (Pakman and Paninski, 2014). From Figures 3.2(b) and 3.2(c), we observe that including the inequality constraints in the conditional distribution provides smaller confidence intervals compared to the ones given by the unconstrained GP. However, they do not satisfy both the boundedness and monotonicity conditions exhibited by the function y. On the other hand, from Figure 3.2(d), imposing both conditions leads to a more accurate prediction and more realistic confidence intervals. Later in subsection 3.3.3, we detail how to obtain the results of Figure 3.2(d).



(a) Unconstrained GP.

(b) GP with boundedness.

(c) GP with monotonicity.

(d) GP with boundedness & monotonicity.

Figure 3.2: Example of Gaussian models satisfying different types of inequality constraints for interpolating the function $x \mapsto \Phi(\frac{x-0.5}{0.2})$. Each panel shows: both training and test points (black dots and red crosses, respectively), the conditional mean function (blue solid line), the 90% confidence interval (grey region), and conditional realisations (dashed lines). For boundedness constraints, horizontal dashed lines denote the bounds.

3.2.3 Connexion between the maximum a posteriori estimate and spline interpolation

In practice, the posterior mode (maximum a posteriori estimate, MAP) of (3.7) can be used as a point estimate of unobserved quantities (Maatouk, 2015; Rasmussen and Williams, 2005). Let $\boldsymbol{\mu}^*$ be the posterior mode that maximises the probability density function (pdf) of $\boldsymbol{\xi}$ conditioned on $\boldsymbol{\Phi\xi} = \boldsymbol{y}$ and $\boldsymbol{\xi} \in \mathcal{C}$. Then, maximising the pdf of $\boldsymbol{\xi}|\{\boldsymbol{\Phi\xi} = \boldsymbol{y}, \boldsymbol{\xi} \in \mathcal{C}\}$ is equivalent to maximise the quadratic problem:

$$\boldsymbol{\mu}^* = \underset{\substack{\boldsymbol{\xi} \text{ s.t. } \boldsymbol{\xi} \in C \\ \boldsymbol{\Phi\xi} = \boldsymbol{y}}}{\arg\max} \{-\boldsymbol{\xi}^\top \boldsymbol{\Gamma}^{-1} \boldsymbol{\xi}\}, \tag{3.8}$$

By maximising (3.8), we are looking for the most likely vector $\boldsymbol{\xi}$ satisfying both interpolation and inequality constraints. The mode in (3.8) has the property that, loosely speaking, it converges to the one provided by spline interpolation (Bay et al., 2016).

Theorem 3.1 (Bay et al. (2016)) *Consider the finite-dimensional representation Y_m in (3.6). Let $Y_m^{MAP}(x)$ bet the MAP estimate given by*

$$Y_m^{MAP}(x) = \sum_{j=1}^{m} \mu_j^* \phi_j(x),$$

with posterior mode $\boldsymbol{\mu}^* = [\mu_1^*, \cdots, \mu_m^*]^\top$ defined in (3.8). Then, Y_m^{MAP} uniformly converges to the spline interpolation under the inequality constraints in $\mathcal{E}$ as $m \to \infty$.

3.2.4 Extension to higher dimensions

The finite-dimensional representation in (3.6) can be extended to d dimensional input spaces by tensorisation. Consider $\boldsymbol{x} = (x_1, \cdots, x_d) \in \mathcal{D}$ with compact input space $\mathcal{D} = [0,1]^d$, and a set of knots per dimension $(t_1^1, \cdots, t_{m_1}^1), \cdots, (t_1^d, \cdots, t_{m_d}^d)$. Then, the finite representation $Y_{m_1,\cdots,m_d}$ is given by

$$Y_{m_1,\cdots,m_d}(\boldsymbol{x}) = \sum_{j_1,\cdots,j_d=1}^{m_1,\cdots,m_d} \left[\prod_{k=1,\cdots,d} \phi_{j_k}^k(x_k) \right] \xi_{j_1,\cdots,j_d}, \quad \text{s.t.} \quad \begin{cases} Y_{m_1,\cdots,m_d}(\boldsymbol{x}_i) = y_i, \\ \boldsymbol{\xi} \in \mathcal{C}, \end{cases} \tag{3.9}$$

where $\boldsymbol{x}_i \in \mathcal{D}$ and $y_i \in \mathbb{R}$, for $i = 1, \cdots, n$, $\xi_{j_1,\cdots,j_d} := Y(t_{j_1}, \cdots, t_{j_d})$ and $\phi_{j_i}^i$ are the hat basis functions defined in (3.2). Following a similar procedure as in subsection 3.2.1, we observe that $\boldsymbol{\xi} = [\xi_{1,\cdots,1}, \ \cdots, \ \xi_{m_1,\cdots,m_d}]^\top$ is a centred Gaussian vector with covariance matrix $\boldsymbol{\Gamma}$ as in (3.7).[1]

3.3 Improvements on the finite-dimensional approximation of Gaussian processes

Now, we extend the framework in Section 3.2 to account for general linear inequality constraints. We also study various efficient Monte Carlo (MC) and Markov Chain Monte Carlo (MCMC) for approximating the posterior distribution of the model.

3.3.1 Extension to general sets of linear inequalities

Here, we continue with the same notation as in Section 3.2. We consider the case where $\mathcal{C}$ is composed by a set of q linear inequalities of the form:

$$\mathcal{C} = \left\{ \mathbf{c} \in \mathbb{R}^m; \ \forall\, k = 1, \ldots, q \ : \ l_k \leq \sum_{j=1}^{m} \lambda_{k,j} c_j \leq u_k \right\}, \tag{3.10}$$

[1] The tensor form of (3.9) has been used in (Maatouk and Bay, 2017) to account only for monotonicity constraints in 2D due to numerical intractability.

where the $\lambda_{k,j}$'s encode the linear operations, the l_k's and u_k's represent the lower and upper bounds. Notice that the sets $\mathcal{C}_\kappa$ in (3.5) are particular cases of $\mathcal{C}$. Denote $\boldsymbol{\Lambda} = (\lambda_{k,j})_{1\leq k\leq q, 1\leq j\leq m}$, $\boldsymbol{l} = (l_k)_{1\leq k\leq q}$, and $\boldsymbol{u} = (u_k)_{1\leq k\leq q}$. Hence, (3.7) is written

$$\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}) \quad \text{s.t.} \quad \begin{cases} \boldsymbol{\Phi}\boldsymbol{\xi} = \boldsymbol{y} & \text{(interpolation conditions)}, \\ \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u} & \text{(inequality conditions)}. \end{cases} \tag{3.11}$$

We further assume that $q \geq m$ and that $\boldsymbol{\Lambda}$ has rank m. By the rank-nullity theorem (Meyer, 2000), this implies that $\boldsymbol{\Lambda}$ is injective. In particular, a linear system of the form $\boldsymbol{\Lambda}\boldsymbol{\xi} = \boldsymbol{\eta}$ admits a unique solution $\boldsymbol{\xi}$ when $\boldsymbol{\eta}$ is in the image space of $\boldsymbol{\Lambda}$. This assumption is verified in many practical situations, up to adding inactive constraints. For instance, the monotonicity condition $\xi_1 \leq \cdots \leq \xi_m$, which involves only $m-1$ (linearly independent) conditions, can be made compatible by adding the condition $-\infty \leq \xi_1$ (and/or $\xi_m \leq \infty$).

We now explain how to sample $\boldsymbol{\xi}$ from (3.11). First, we compute the conditional distribution given the interpolation constraints $\boldsymbol{\xi}|\{\boldsymbol{\Phi}\boldsymbol{\xi} = \boldsymbol{y}\}$. Since $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$, then $\boldsymbol{\Phi}\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^\top)$ and the conditional distribution $\boldsymbol{\xi}|\{\boldsymbol{\Phi}\boldsymbol{\xi} = \boldsymbol{y}\}$ is also Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with conditional parameters given by

$$\boldsymbol{\mu} = \boldsymbol{\Gamma}\boldsymbol{\Phi}^\top[\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^\top]^{-1}\boldsymbol{y}, \quad \text{and} \quad \boldsymbol{\Sigma} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma}\boldsymbol{\Phi}^\top[\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^\top]^{-1}\boldsymbol{\Phi}\boldsymbol{\Gamma}. \tag{3.12}$$

Therefore, we have $\boldsymbol{\Lambda}\boldsymbol{\xi}|\{\boldsymbol{\Phi}\boldsymbol{\xi} = \boldsymbol{y}\} \sim \mathcal{N}(\boldsymbol{\Lambda}\boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Sigma}\boldsymbol{\Lambda}^\top)$. Let $\mathcal{TN}(\mathbf{m}, \mathbf{C}, \mathbf{a}, \mathbf{b})$ be the truncated multinormal distribution with mean vector $\mathbf{m}$, covariance matrix $\mathbf{C}$, and the bound vectors $(\mathbf{a}, \mathbf{b})$ such that $\mathbf{a} \leq \mathbf{b}$. Thus, the posterior of (3.11) is obtained from

$$\boldsymbol{\Lambda}\boldsymbol{\xi}|\{\boldsymbol{\Phi}\boldsymbol{\xi} = \boldsymbol{y}, \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u}\} \sim \mathcal{TN}(\boldsymbol{\Lambda}\boldsymbol{\mu},\ \boldsymbol{\Lambda}\boldsymbol{\Sigma}\boldsymbol{\Lambda}^\top,\ \boldsymbol{l},\ \boldsymbol{u}). \tag{3.13}$$

Notice that the inequality conditions are encoded in the posterior mean $\boldsymbol{\Lambda}\boldsymbol{\mu}$, the posterior covariance $\boldsymbol{\Lambda}\boldsymbol{\Sigma}\boldsymbol{\Lambda}^\top$, and the bounds $(\boldsymbol{l}, \boldsymbol{u})$. The truncated multinormal in (3.13) can be approximated using the MC/MCMC algorithms described in subsection 3.3.2. Denoting $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\xi}$, notice that samples for $\boldsymbol{\xi}$ can be obtained by sampling from $\boldsymbol{\eta}$ and solving a linear system. Indeed, as mentioned above, we assumed that $\boldsymbol{\Lambda}$ has rank m, which implies that the solution of $\boldsymbol{\Lambda}\boldsymbol{\xi} = \boldsymbol{\eta}$ exists and is unique. The whole sampling scheme is summarised in Algorithm 1.

From Algorithm 1, note that the computation of the posterior mode is required for sampling purposes. Using (3.13), the quadratic problem in (3.8) can be written as

$$\boldsymbol{\mu}^* = \underset{\boldsymbol{\xi} \text{ s.t. } \boldsymbol{l}\leq\boldsymbol{\Lambda}\boldsymbol{\xi}\leq\boldsymbol{u}}{\arg\max} \{-[\boldsymbol{\xi} - \boldsymbol{\mu}]^\top\boldsymbol{\Sigma}^{-1}[\boldsymbol{\xi} - \boldsymbol{\mu}]\}, \tag{3.14}$$

with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ as in (3.12). In contrast to (3.8), here the interpolation constraints are directly encoded in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Then, the optimisation problem in (3.14) is equivalent to

$$\boldsymbol{\mu}^* = \underset{\boldsymbol{\xi} \text{ s.t. } \boldsymbol{l}\leq\boldsymbol{\Lambda}\boldsymbol{\xi}\leq\boldsymbol{u}}{\arg\min} \{\boldsymbol{\xi}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi} - 2\boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}\}, \tag{3.15}$$

which can be solved via quadratic programming (Goldfarb and Idnani, 1982). Finally, the posterior mode of (3.13) is given by $\boldsymbol{\nu}^* = \boldsymbol{\Lambda}\boldsymbol{\mu}^*$ with $\boldsymbol{\mu}^*$ as in (3.15).

Algorithm 1 Sampling from the finite-dimensional GP with linear inequality constraints.

Procedure: Sampling from $\boldsymbol{\xi}|\{\boldsymbol{\Phi\xi} = \boldsymbol{y}, \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u}\}$, where $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma})$
Input: $\boldsymbol{y}$, $\boldsymbol{\Gamma} \in \mathbb{R}^{m\times m}$, $\boldsymbol{\Phi} \in \mathbb{R}^{n\times m}$, $\boldsymbol{\Lambda}$, $\boldsymbol{l}$, $\boldsymbol{u}$.
Compute the conditional mean and covariance of $\boldsymbol{\xi}|\{\boldsymbol{\Phi\xi} = \boldsymbol{y}\}$:
    $\boldsymbol{\mu} = \boldsymbol{\Gamma\Phi}^\top(\boldsymbol{\Phi\Gamma\Phi}^\top)^{-1}\boldsymbol{y}$,
    $\boldsymbol{\Sigma} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma\Phi}^\top(\boldsymbol{\Phi\Gamma\Phi}^\top)^{-1}\boldsymbol{\Phi\Gamma}$.
Solve the quadratic problem in $\mathbb{R}^m$: $\boldsymbol{\mu}^* = \min_{\boldsymbol{\xi}\in\mathbb{R}^m}\{\boldsymbol{\xi}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{\xi}|\ \boldsymbol{\Phi\xi} = \boldsymbol{y}, \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u}\}$.
Sample from the truncated multinormal distribution according to subsection 3.3.2:
    $\boldsymbol{\Lambda\xi}|\{\boldsymbol{\Phi\xi} = \boldsymbol{y}, \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u}\} \sim \mathcal{TN}\left(\boldsymbol{\Lambda\mu},\ \boldsymbol{\Lambda\Sigma\Lambda}^\top,\ \boldsymbol{l},\ \boldsymbol{u}\right)$.
Define $\boldsymbol{\eta} = \boldsymbol{\Lambda\xi}$, and solve the linear system to obtain the sample $\boldsymbol{\xi}$.
Remark: use the posterior mode $\boldsymbol{\nu}^* = \boldsymbol{\Lambda\mu}^*$ as a starting state for MCMC samplers.

3.3.2 Efficient sampling from the posterior distribution

As shown in (3.13), the posterior distribution $\boldsymbol{\Lambda\xi}|\{\boldsymbol{\Phi\xi} = \boldsymbol{y}, \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u}\}$ is truncated multinormal. It is supported on $\mathbb{R}^q$, where $q \geq m$ is defined in (3.10). Notice that m should be chosen large enough to get good approximations. An MC algorithm based on rejection sampling was proposed in (Maatouk and Bay, 2016) using the posterior mode. This method, called rejection sampling from the mode (RSM), is an exact sampler that provides independent and identically distributed (iid) sample paths. However, the acceptance rate from RSM decreases when m gets larger, providing a poor performance for high dimensional spaces. Recently, another MC-based exact sampler using the separation-of-variables technique from (Genz, 1992) was introduced by Botev (2017) to deal with truncated multinormals in high dimensions. As in (Genz, 1992), the approach in (Botev, 2017) can both sample Gaussian random variables under linear inequality constraints, and estimate the probabilities that these constraints are satisfied, via minimax exponential tilting (ET). Since RSM and ET are exact methods, we use them as gold standards to evaluate the performance of the MCMC techniques that we describe now.

MCMC approaches use a Markov chain to sample the posterior distribution, providing correlated samples but with potentially higher acceptance rates. Efficient algorithms have been proposed for truncated multinormal distributions such as Gibbs sampling (Taylor and Benjamini, 2016), Metropolis-Hastings (Murphy, 2012), and Hamiltonian Monte Carlo (HMC) (Pakman and Paninski, 2014). In this section, we apply them to sample from the distribution of (3.13).

Gibbs sampling. Algorithms based on Gibbs sampling are widely used to sample from truncated multinormals due to their easy implementation and their reliable performances (Brooks et al., 2011; Murphy, 2012). They sample each variable in turn conditional on the values of the other ones. Therefore, sampling from a truncated multinormal is reduced to sampling sequentially from conditional truncated (univariate) normals. Unlike RSM, there is no rejection step. However, the "single site updating" property may produce strong correlations, requiring to discard intermediate samples (thinning effect). In this chapter, we use the fast Gibbs sampler proposed in (Taylor and Benjamini, 2016).

Metropolis Hastings (MH). MH-based algorithms propose to move all the coordinates at a time in each step to obtain less correlated simulations. For multinormal distributions, a symmetric Gaussian proposal is commonly used, i.e. $q(\boldsymbol{x}'|\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x}, \eta\boldsymbol{\Sigma}_{\boldsymbol{x}'|\boldsymbol{x}}\right)$ where η is a scale factor. The proposed state $\boldsymbol{x}'$ is either accepted or rejected according to a given acceptance rule (Murphy, 2012). If the proposal is accepted, the new state is $\boldsymbol{x}'$, otherwise the new state remains at the previous state $\boldsymbol{x}$. This approach is known as the random walk Metropolis algorithm. One can increase the acceptance rate by tuning properly the value of η. Here, we assume that $\boldsymbol{\Sigma}_{\boldsymbol{x}'|\boldsymbol{x}}$ is given by the covariance function of the posterior we want to approximate, i.e. $\boldsymbol{\Sigma}_{\boldsymbol{x}'|\boldsymbol{x}} = \boldsymbol{\Lambda}\boldsymbol{\Sigma}\boldsymbol{\Lambda}^\top$ with $\boldsymbol{\Sigma}$ defined as in (3.12).

Hamiltonian Monte Carlo (HMC). Hybrid methods have been subject to great attention from the statistical community due to the inclusion of physical interpretation that may provide useful intuitions (Brooks et al., 2011). In (Duane et al., 1987), an efficient hybrid approach was introduced using the properties of Hamiltonian dynamics. Later in (Neal, 1996), the hybrid approach from (Duane et al., 1987) was extended to statistical applications, and was introduced formally as HMC. The Hamiltonian dynamics provide distant proposal distributions producing less correlated sample paths without diminishing the acceptance rate. In this manuscript, we use the HMC-based approach for truncated multinormals introduced in (Pakman and Paninski, 2014), which follows the same dynamics as a classical HMC sampler, but the particle "bounces" on the boundaries if its trajectory reaches one of the inequality constraints.

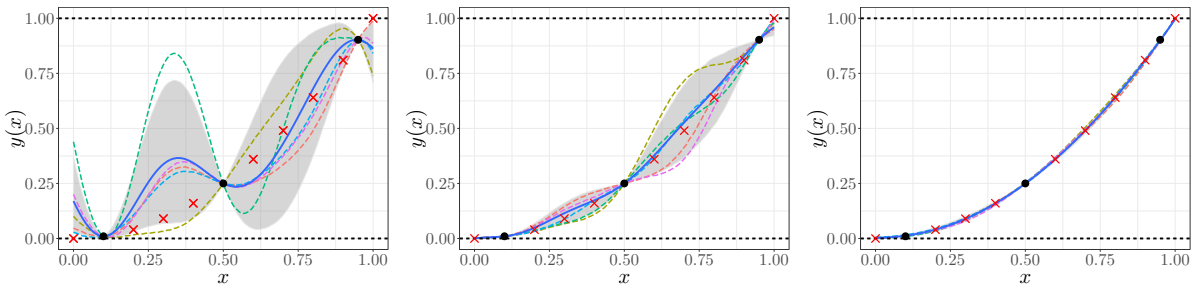
3.3.3 Numerical illustrations

We now show on synthetic examples that the proposed framework enables us to address different types of constraints in 1D and 2D. For instance, the number of knots m are manually fixed aiming a trade-off between high quality of resolution and computational cost. We refer to Chapter 5 for a discussion on a sequential algorithm for the automatic knot insertion in regions requiring a high quality of resolution (e.g., in highly variable regions).

3.3.3.1 Illustration on 1D examples

Here, we fix $m = 100$ and approximate the posterior distribution of (3.13) via HMC. In the three toy examples, we use a SE kernel with parameters ($\sigma^2 = 1.0,\ \ell = 0.2$).

Example 1. We continue with the example in Figure 3.1. As we can fix the structure of the linear inequalities $(\boldsymbol{\Lambda}, \boldsymbol{l}, \boldsymbol{u})$, we can impose both boundedness and monotonicity conditions in the constrained GP. To do so, one way is to encode them individually. Let $\boldsymbol{l}_1 \leq \boldsymbol{\Lambda}_1\boldsymbol{\xi} \leq \boldsymbol{u}_1$ and $\boldsymbol{l}_2 \leq \boldsymbol{\Lambda}_2\boldsymbol{\xi} \leq \boldsymbol{u}_2$ be the sets of conditions to satisfy boundedness and monotonicity constraints, respectively. Then, we can build an extended set of inequalities $\boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u}$ by stacking the constraints (i.e. $\boldsymbol{\Lambda} = [\boldsymbol{\Lambda}_1^\top, \boldsymbol{\Lambda}_2^\top]^\top, \boldsymbol{l} = [\boldsymbol{l}_1^\top, \boldsymbol{l}_2^\top]^\top, \boldsymbol{u} = [\boldsymbol{u}_1^\top, \boldsymbol{u}_2^\top]^\top$). Notice that one can encode the same information in a reduced set of linear inequalities. Instead of encoding independently both constraints, which requires $q =$



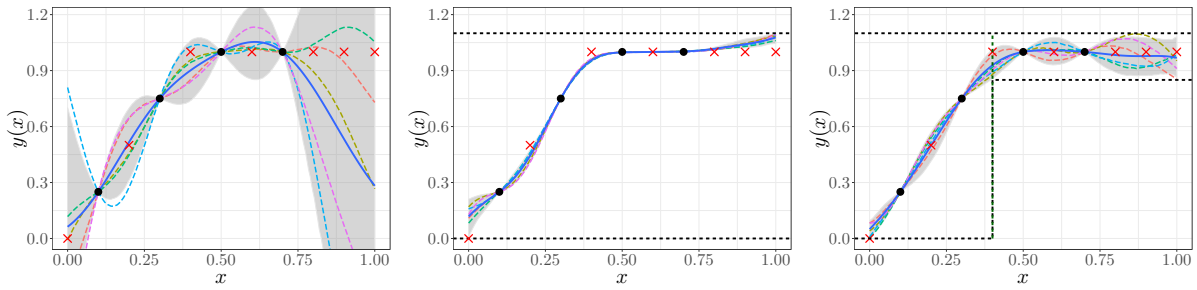
(a) GP under boundedness constraints. (b) Adding monotonicity to GP in (a). (c) Adding convexity to GP in (b).

Figure 3.3: Gaussian models satisfying one or several types of inequality constraints for interpolating the square function $x \mapsto x^2$. Panel description is the same as in Figure 3.2.

$2m-1$ inequalities, one can impose boundedness conditions only for the first and last knot, and monotonicity conditions for all the knots except the first one. Due to monotonicity, the intermediate knots will also satisfy the boundaries. In this case, we only need to impose $q = m+1$ conditions. In many other cases, the size of specific sets of constraints can be reduced. However for general discussions, we use the full extended set and apply efficient samplers to approximate the posterior.

Example 2. Note from the previous example that the extension to more than two sets of inequalities is straightforward. Consider for instance Q different sets of conditions. We can build the posterior distribution in (3.13) with $\boldsymbol{\Lambda} = [\boldsymbol{\Lambda}_1^\top, \cdots, \boldsymbol{\Lambda}_Q^\top]^\top$, $\boldsymbol{l} = [\boldsymbol{l}_1^\top, \cdots, \boldsymbol{l}_Q^\top]^\top$, and $\boldsymbol{u} = [\boldsymbol{u}_1^\top, \cdots, \boldsymbol{u}_Q^\top]^\top$, and apply Algorithm 1. Figure 3.3 shows an example with the target function $y(x) = x^2$, satisfying three types of inequality constraints: boundedness, monotonicity and convexity. We proposed different models satisfying one or more inequality constraints. Observe that, by imposing the three conditions, we obtain samples that also satisfy the three types of constraints.

Example 3. Since the bounds $(\boldsymbol{l}, \boldsymbol{u})$ are not forced to be the same everywhere, it is possible to fix specific constraints over non-overlapping intervals. For instance if the interval is partitioned into G subintervals, we consider the corresponding partition $\boldsymbol{\xi} = [\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_G]^\top$. Then, we can impose different types of inequality conditions in each group by considering the same structure used in the previous example. Figure 3.4 shows an example where the function y satisfies different behaviours in two non-overlapping intervals. The output increases monotonically and peaks at $y(0.4) = 1.0$. This kind of profile is met in different applications (e.g. step responses in control theory, protein profiles in biology, Kocijan, 2016; Murphy, 2012). We trained three types of models satisfying different conditions. For the case of multiple constraints, we imposed both boundedness and monotonicity. For the case of sequential conditions, we divided the profile in two non-overlapping intervals satisfying different types of constraints. Note that, by imposing sequentially the constraints, we obtain less restricted uncertainties and more accurate models for data fitting.



(a) Unconstrained GP. (b) GP with multiple constraints. (c) GP with sequential constraints.

Figure 3.4: Gaussian models with different types of constraints for the example 3 from subsection 3.3.3. (a) Unconstrained GP. (b) Boundedness and monotonicity constraints are imposed. (c) The two non-overlapping intervals are divided by a vertical dashed line at 0.4. In the first interval, boundedness and monotonicity constraints are taken into account. In the second interval, only boundedness is imposed.

3.3.3.2 Performance of MC and MCMC samplers

In Table 3.1, we evaluate the efficiency of the MC and MCMC approaches described in subsection 3.3.2 on the examples in Figure 3.2. In order to reduce the sampling cost, we used $m = 30$ hat basis functions. Hence, the problem is to sample a vector of length 30 from a truncated multinormal distribution. Note from Algorithm 1 that the posterior mode is used as the starting state for the MCMC samplers. This results in chains that are initialised into high probability regions, and only a small amount of simulations have to be "burned" in order to obtain samples that are appear to be independent of the initialisation location. Therefore, we only burned the first 100 simulations from all the MCMC samplers. We set the tuning hyperparameters such that the effective sample size (ESS) is within the ranges produced by both RSM and ET (grey columns). The ESS is a heuristic used commonly to evaluate the quality of correlated sample paths, and it gives an intuition on how many samples from the path can be considered independent (Gong and Flegal, 2016). A standard ESS is given by $\text{ESS} = n_s/(1 + 2\sum_{k=1}^{n_s} \rho_k)$ where n_s is the size of the sample path and ρ_k is the sample autocorrelation with lag k. However, the drawback of this indicator is that it accepts negative correlations to evaluate the quality of mean estimators (e.g. for variance reduction). Thus, we suggest to use the initial convex sequence estimator proposed by Geyer (1992) in order to obtain non-negative, non-increasing and convex sample autocorrelations $\widehat{\rho}_k$. Then, we obtain $\text{ESS} = n_s/(1 + 2\sum_{k=1}^{n_s} \widehat{\rho}_k)$. We compute the ESS indicator for each coordinate of $\boldsymbol{\xi}$, i.e. $\text{ESS}_j = \text{ESS}(\xi_j^1, \cdots, \xi_j^{n_s})$ for the j-th component of $\boldsymbol{\xi}$ with $j = 1, \cdots, m$. We then compute quantiles $(q_{10\%}, q_{50\%}, q_{90\%})$ over the 30 resulting ESS values. To take into account cross-correlations from multivariate MCMC, we also compute the multivariate ESS (mvESS) proposed by Vats et al. (2017). For the mvESS, values higher than n_s indicate the presence of negative correlations. In our case, we are interested in being around n_s. The size $n_s = 10^4$ is chosen to be larger than the minimum ESS required to obtain a proper

Table 3.1: Efficiency of MC and MCMC samplers in subsection 3.3.2 (by rows) in terms of ESS-based indicators (by columns).

Toy Example	Sampler	CPU Time $[s]$	ESS $[\times 10^4]$ $(q_{10\%}, q_{50\%}, q_{90\%})$	mvESS $[\times 10^4 s^{-1}]$	TN-ESS $[\times 10^4 s^{-1}]$	Hyperparameter
Toy Figure 3.2(b) (Boundedness)	RSM	99.06	(0.93, 0.98, 1.00)	1.22	0.01	-
	ET	**0.44**	(0.94, 0.97, 1.00)	1.17	**2.14**	-
	Gibbs	3.54	(0.95, 0.98, 1.00)	1.16	0.27	thinning = 200
	MH	52.21	(0.98, 1.00, 1.00)	1.21	0.02	$\eta = 1$
	HMC	**0.44**	(0.90, 0.93, 1.00)	1.26	2.04	-
Toy Figure 3.2(c) (Monotonicity)	RSM	190.62	(1.00, 1.00, 1.00)	1.21	0.01	-
	ET	0.77	(0.95, 0.98, 1.00)	1.18	1.23	-
	Gibbs	3.04	(1.00, 1.00, 1.00)	1.15	0.33	thinning = 200
	MH	96.64	(0.98, 1.00, 1.00)	1.23	0.01	$\eta = 1$
	HMC	**0.33**	(0.89, 0.94, 0.98)	1.28	**2.70**	-
Toy Figure 3.2(d) (Bounded Monotonicity)	RSM	-	-	-	-	-
	ET	41.16	(0.99, 1.00, 1.00)	1.23	0.02	-
	Gibbs	40.28	(0.37, 0.60, 0.91)	1.09	0.01	thinning = 1000
	MH	-	-	-	-	-
	HMC	**12.92**	(0.85, 0.93, 1.00)	1.26	**0.07**	-

estimation of the vector $\boldsymbol{\xi} \in \mathbb{R}^{30}$: minESS(30) = 8563 (Gong and Flegal, 2016; Vats et al., 2017). Finally, using the procedure proposed in (Lan and Shahbaba, 2016), we test the efficiency of each method by computing the time normalised ESS (TN-ESS) at $q_{10\%}$ (worst case) using the CPU time in seconds, i.e. TN-ESS = $q_{10\%}$(ESS)/(CPU Time).

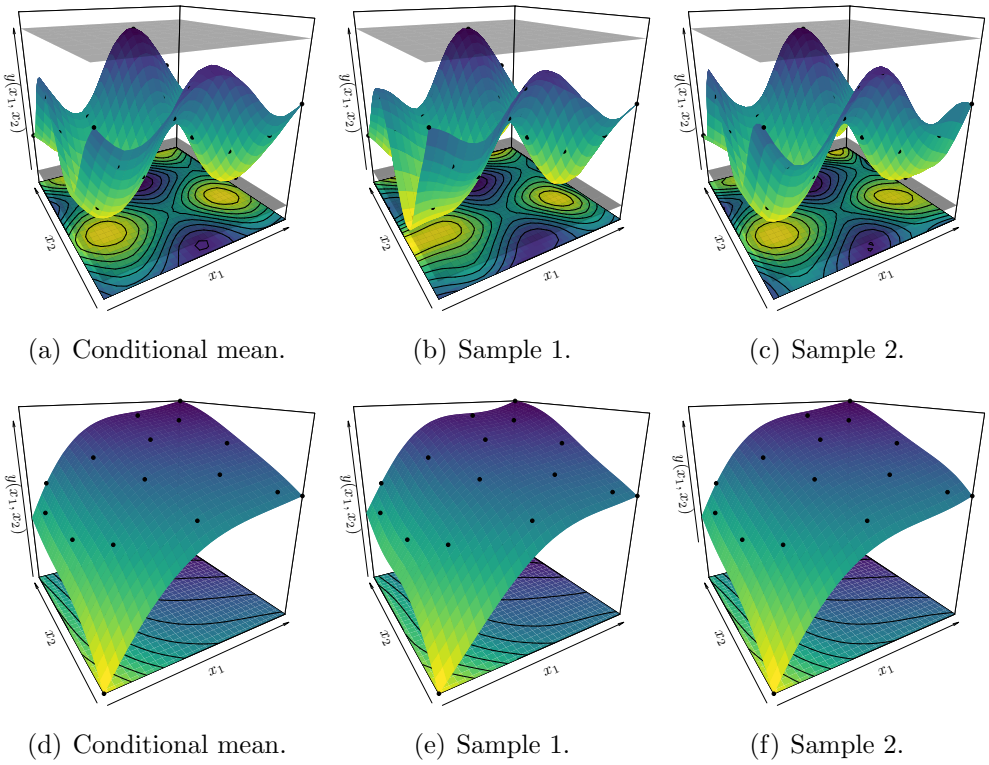
Note from Table 3.1 that, for both examples in Figures 3.2(b) and 3.2(c), the MC/MCMC techniques tend to produce similar ESS intervals, but RSM and MH are the most expensive procedures due to their high rejection rates. Although the Gibbs sampler requires to discard a large amount of simulations in order to be within reasonable ESS ranges, it also yields accurate results in both efficiency and CPU time. In general, both ET and HMC methods yield more efficient results than the other samplers in the first two examples. For more complex constraints as in the example of Figure 3.2(d), the efficiency is reduced dramatically for all the methods. For example, the acceptance rates of both RSM and MH are so small that sampling was not feasible. For the other methods, the TN-ESS rates are smaller but HMC still gives a reasonable value (three times larger than for ET), which leads us to conclude that HMC is an efficient sampler for the proposed framework.

3.3.3.3 Illustration on 2D examples

Let the 2D finite approximation be given by

$$Y_{m_1,m_2}(x_1,x_2) := \sum_{j_1=1}^{m_1}\sum_{j_2=1}^{m_2} \phi^1_{j_1}(x_1)\phi^2_{j_2}(x_2)\xi_{j_1,j_2}, \quad \text{s.t.} \quad \begin{cases} Y_{m_1,m_2}(x_1^i, x_2^i) = y_i, \\ \boldsymbol{\xi} \in \mathcal{C}, \end{cases} \tag{3.16}$$

where $\xi_{j_1,j_2} = Y(t_{j_1}, t_{j_2})$, $\phi^1_{j_1}$ and $\phi^2_{j_2}$ are hat basis functions given by (3.2), and (x_1^i, x_2^i), for $i = 1, \cdots, n$, denote a DoE. Consider the centred Gaussian vector



(a) Conditional mean. (b) Sample 1. (c) Sample 2.

(d) Conditional mean. (e) Sample 1. (f) Sample 2.

Figure 3.5: Examples of 2D Gaussian models with boundedness or monotonicity constraints for interpolating the toy examples from subsection 3.3.3.3. Boundedness and monotonicity results are shown in the first and second row, respectively. Each row shows: training points (black dots), the conditional mean function (on the left), and some conditional samples (last two columns on the right). For boundedness constraints, (horizontal) grey surfaces denote the bounds.

$\boldsymbol{\xi} = [\xi_{1,1}, \cdots, \xi_{1,m_2}, \cdots, \xi_{m_1,1}, \cdots, \xi_{m_1,m_2}]^\top$ with covariance matrix $\boldsymbol{\Gamma}$. Notice that each row of the matrix $\boldsymbol{\Phi}$ is given by

$$\boldsymbol{\Phi}_{i,:} = \left[\phi_1^1(x_1^i)\phi_1^2(x_2^i) \quad \cdots \quad \phi_1^1(x_1^i)\phi_{m_2}^2(x_2^i) \quad \cdots \quad \phi_{m_1}^1(x_1^i)\phi_1^2(x_2^i) \quad \cdots \quad \phi_{m_1}^1(x_1^i)\phi_{m_2}^2(x_2^i)\right],$$

for $i = 1, \cdots, n$. Finally, the posterior distribution of (3.16) can be computed, and the routine follows Algorithm 1.

Figure 3.5 shows 2D examples with boundedness or monotonicity constraints. We use a 2D SE kernel and estimate the covariance parameters via ML.[2] The training points are generated with a maximin Latin hypercube DoE over $[0,1]^2$.[3] The functions are: 3.5(a) $y(x_1, x_2) = -\frac{1}{2}[\sin(9x_1) - \cos(9x_2)]$, and 3.5(d) $y(x_1, x_2) = \arctan(5x_1) + \arctan(x_2)$.

[2] 2D SE kernel: $k_{\boldsymbol{\theta}}(\boldsymbol{x} - \boldsymbol{x}') = \sigma^2 \exp\left\{-\frac{(x_1 - x_1')^2}{2\ell_1^2} - \frac{(x_2 - x_2')^2}{2\ell_2^2}\right\}$ with $\boldsymbol{\theta} = (\sigma^2, \ell_1, \ell_2)$.

[3] A maximin Latin hypercube DoE is a space-filling design consisting in the iterative maximisation of the distance between two closest design points from a random Latin hypercube design. Here, we used the simulated annealing routine (`maximinSA_LHS`) from the R package `DiceDesign` (Dupuy et al., 2015).

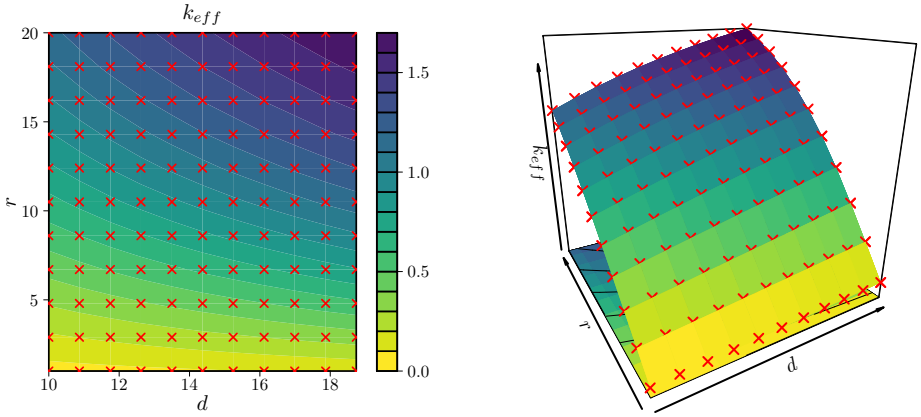


Figure 3.6: Nuclear criticality safety assessment. (Left) 2D visualisation of the k_{eff} values measured over a regular grid. (Right) 3D visualisation of the k_{eff} data.

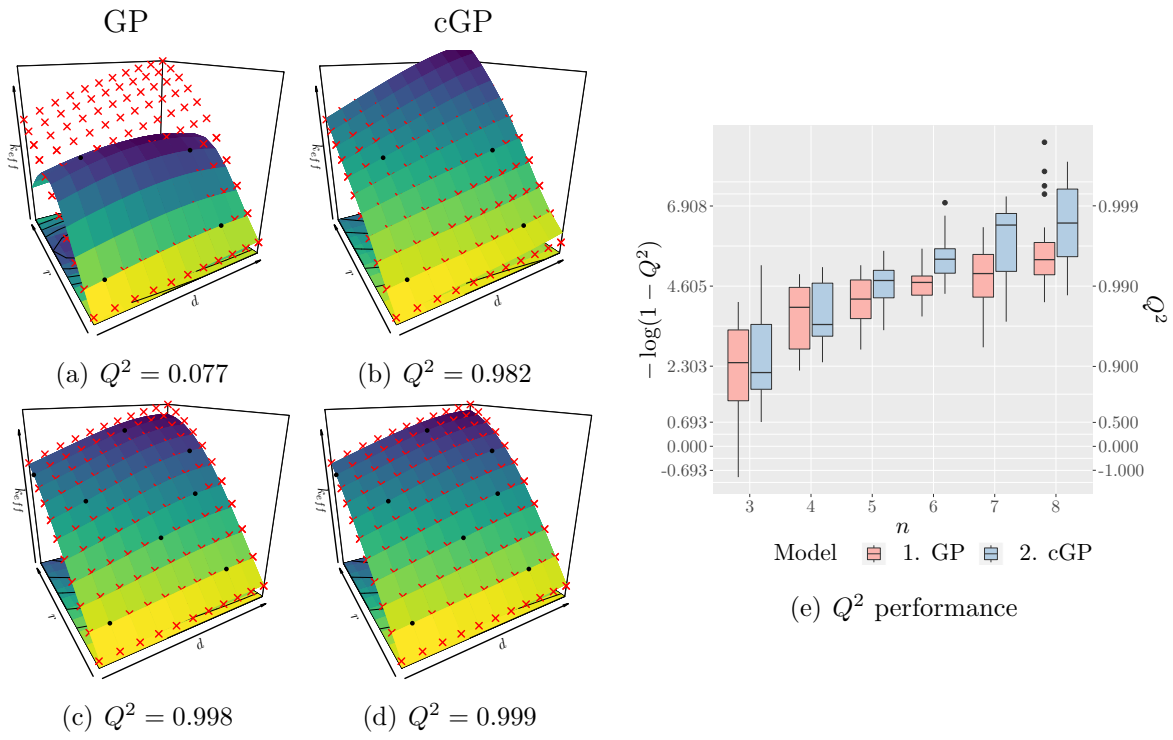
For the case of monotonicity in 2D, the constraints to be satisfied are: $\xi_{i+1,j} \geq \xi_{i,j}$ and $\xi_{i,j+1} \geq \xi_{i,j}$, for $i = 1, \cdots, m_1 - 1$ and $j = 1, \cdots, m_2 - 1$. This means that the function is non-decreasing w.r.t. its two input variables.

3.4 2D application: nuclear safety criticality

For assessing the stability of neutron production in nuclear reactors, safety criteria based on the effective neutron multiplication factor k_{eff} are commonly used (IAE, 2014; Fernex et al., 2005). This factor is defined as the ratio of the total number of neutrons produced by a fission chain reaction to the total number of neutrons lost by absorption and leakage. Besides the geometry and composition of fissile materials (e.g. mass, density), the k_{eff} is sensitive to other types of parameters like the structure materials characteristics (e.g. concrete), and the presence of specific materials (e.g. moderators). Since the optimal control of an individual parameter or a combination of them can lead to safe conditions, the understanding of their influence in criticality safety assessment is crucial.

Here, we applied the proposed framework to a dataset provided by the IRSN ("Institut de Radioprotection et de Sûreté Nucléaire", in French). The k_{eff} factor was obtained from a nuclear reactor called "Lady Godiva device" originally situated at "Los Alamos National Laboratory", New Mexico, U.S. Two parameters of uranium spheres are considered: the radius r and density d. The dataset contains 121 observations in a 11×11 grid (see Figure 3.6). Notice that, on the domain considered for the input variables, the k_{eff} increases as the radius and density of the uranium sphere increase.

We trained different Gaussian models whether the inequality constraints are considered or not. For all the models, the input space was normalised to $[0, 1]^2$, and we used the same 2D SE kernels as for the example from Figure 3.5. We estimated the covariance parameters via ML in the ranges $\sigma^2 \in [0.2, 1]$ and $\ell_1, \ell_2 \in [0.1, 0.9]$. For the constrained model, since the k_{eff} factor indicates the production rate of neutron population, the



(a) $Q^2 = 0.077$ (b) $Q^2 = 0.982$

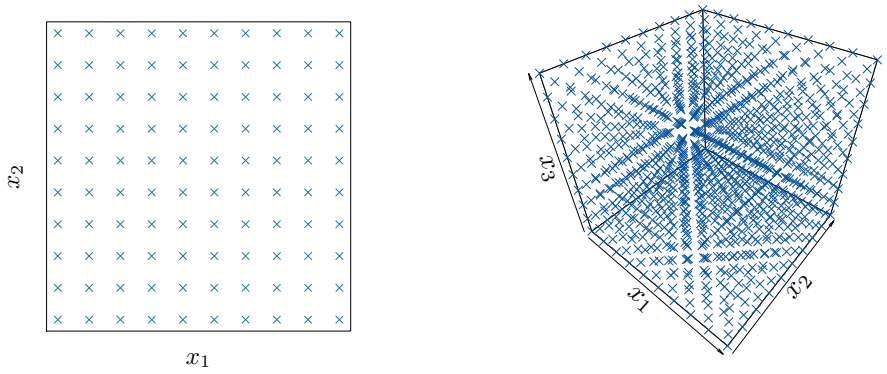
(c) $Q^2 = 0.998$ (d) $Q^2 = 0.999$

(e) Q^2 performance

Figure 3.7: 2D Gaussian models for interpolating the Godiva's dataset. Unconstrained GP models are trained using MLE (left column) and using (a) four, or (c) eight training points n from the proposed maximin Latin hypercube DoE. Constrained GP (cGP) models are also trained using MLE (middle column). Each panel shows: training and test points (black dots and red crosses), and the conditional mean function (solid surface). (e) Q^2 assessment of the Gaussian models using various number of training points n and using twenty different random Latin hypercube designs. Results are shown for the unconstrained GP (red) and cGP (blue).

output of the constrained process has to be positive. In order to take into account non-decreasing behaviours, we also consider monotonicity constraints. We trained both unconstrained and constrained models with a fixed maximin Latin hypercube DoE at eight locations extracted from the unit grid. We used the remaining data to assess the quality of prediction tasks.

Figure 3.7 shows the performance of the proposed models using four or eight points from the proposed fixed DoE. For the unconstrained model, we observe that the quality of the predictions depends strongly on both the amount of training data and their distribution in the input space. Notice from Figure 3.7(a) that if only few training points are available, predictions are poor and they do not satisfy positive and non-decreasing behaviours. In Figure 3.7(c), we observe that if there are enough training data that cover the input space, the unconstrained model behaves well and provides reliable predictions. On the other hand, we observe that the constrained model produce accurate prediction results also when the training set is small.



(a) $d = 2,\ m = 10^2$ (b) $d = 3,\ m = 10^3$

Figure 3.8: 2D and 3D visualisations of equispaced designs using ten knots per dimension.

Because the prediction accuracy depends on the training set, we repeated the procedure with twenty different random Latin hypercube DoEs using several values of n. We used the Q^2 criterion to evaluate the quality of the predictions. Denoting by n_t the number of test points, $z_1, \cdots, z_{n_t}$ and $\widehat{z}_1, \cdots, \widehat{z}_{n_t}$ the sets of test and predicted observations (respectively), then

$$Q^2 = 1 - \frac{\sum_{i=1}^{n_t} (\widehat{z}_i - z_i)^2}{\sum_{i=1}^{n_t} (\overline{z} - z_i)^2}, \tag{3.17}$$

where $\overline{z}$ is the average of the test data. For noise-free observations, the Q^2 is equal to one if the predictors $\widehat{z}_1, \cdots, \widehat{z}_{n_t}$ are exactly equal to the test data (ideal case), zero if they are equal to the constant prediction $\overline{z}$, and negative if they perform worse than $\overline{z}$. Figure 3.7(e) shows that the constrained GP often outperformed the unconstrained one.

3.5 Curse of dimensionality

The number of terms in the tensor construction in (3.9) increases exponentially with the dimension d (see, e.g., Figure 3.8). The computational intractability is led by the approximation of the posterior of (3.9) via MC/MCMC. Since satisfying the interpolation conditions requires having more knots than observations, then MC/MCMC samplers are commonly performed in high dimensions with potential highly restricted sample spaces. More precisely, the computational complexity of the exact HMC sampler in (Pakman and Paninski, 2014) scales linearly with the number of inequality conditions if iterations do not require any reflection, but also increases with each bounce. Hence, in the best case, and assuming $m_1 = \cdots = m_d = m$, the computational complexity is $\mathcal{O}(m^d)$ (e.g. positiveness constraints).

This drawback could be mitigated in different ways. First, one may relax the interpolation of observations through a noise effect. In that case, we can have less knots than observations, and MC/MCMC methods can be applied in lower dimensional spaces.

Then, the proposed framework would lead to less expensive procedures since the cost of the MC and MCMC procedures grows with the number of knots (see Chapter 4). Second, the knots can be placed preferentially in highly variable regions while preserving the tensor-based structure. This is shown in Chapter 5 using an algorithm for the sequential knot insertion in rectangular designs. Third, supplementary assumptions on the nature of the target function can also be made to reduce the dimensionality of the sample spaces where samplers are performed (e.g. considering inactive input variables or additivity). We refer to Chapter 6 for a discussion when additive functions are considered. Finally, we believe that using other types of designs for the knots, that scale better to high dimensions, could have a great impact in the efficiency of the proposed model. To do so, in Chapter 10, we explore basis functions based on Delaunay triangulations that alleviate the grid constraint for the knots.

3.6 Conclusions

We have investigated the extension of the GP-based framework in (Maatouk and Bay, 2017) for general linear inequality constraints. The proposed finite-dimensional approximation takes into account the inequalities for both data interpolation and uncertainty quantification. Since the posterior distribution, conditioned to both interpolation and inequality constraints, is expressed as a truncated multinormal distribution, the exact HMC proposed by Pakman and Paninski (2014) adapts to our needs for sampling purposes. In contrast to previous attempts in (Maatouk and Bay, 2017), we demonstrate on various experimental results that the proposed framework together with the HMC sampler can be easily applied in practice. We refer to Chapter 9 for further details on numerical implementations in R programming language (R Core Team, 2018).

We tested the model on both synthetic and real-world data in 1D and 2D. According to experimental results under different types of inequalities, the proposed framework fits properly the observations, and it provides more realistic confidence intervals compared to unconstrained approaches. It is also flexible to account for multiple sets of inequality conditions (e.g. both positivity and monotonicity) or more complex linear constraints. Moreover, as shown in a 2D nuclear application, the framework provides reliable predictions satisfying both positivity and monotonicity constraints exhibited by the effective neutron multiplication factor.

As discussed throughout this chapter, the proposed framework still presents some limitations in high dimensions due to the tensor-based structure of (3.9). Its efficiency depends on the number of knots used in the finite approximation. For higher values of m, the quality of resolution is better but at the cost of more expensive sampling procedures (see Section 3.5 for a further discussion). Moreover, having robust GP-based models when the number of observations is large is also of interest. In Chapters 4 to 6, we focus on addressing these limitations in order to scale the approach when the input space involves hundreds of input variables and/or thousands of observations.

Chapter 4

Constrained Gaussian Processes with Noisy Observations

Contents

4.1 Introduction

As discussed in Chapter 3, the resulting posterior conditioned to both interpolation and inequality constraints is a truncated GP, therefore, its distribution cannot be computed in closed-form. We demonstrated on both synthetic and real-world examples that the proposed framework together with a HMC sampler provides efficient results on both data fitting and uncertainty quantification. However, strictly interpolating the observations may entail expensive computations due to highly restrictive sample spaces. Moreover, since the condition of having more knots than observations is required (i.e. $m \geq n$), the computational complexity of the HMC sampler becomes more expensive as n increases.

Starting from the claim that allowing noisy observations could yield less constrained sample spaces for MC/MCMC samplers, in this chapter we develop the corresponding approximation of constrained GP models when adding a noise effect. Since we can have less knots than observations, both MC and MCMC algorithms can be applied in lower di-

mensional spaces. Thus, the proposed framework would lead to less expensive procedures since the cost of samplers grows with the number of knots. Moreover, constrained models for observations that are truly noisy are also of interest for practical implementations.

We test the efficiency of various MC and MCMC samplers under 1D toy examples where models without observation noise yield impractical sampling routines. We also show that, on a 5D monotonic example, the framework can be used for thousands of observations, providing high-quality effective sample sizes within reasonable running times. Finally, on 2D and 5D coastal flooding applications, we demonstrate that more flexible and realistic GP implementations can be obtained by considering noise effects and by enforcing the (linear) inequality constraints.

This chapter is based on the preprint:

- López-Lopera, A. F., Bachoc, F., Durrande, N., Rohmer, J., Idier, D., and Roustant O. (2018). Approximating Gaussian process emulators with linear inequality constraints and noisy observations via MC and MCMC. ArXiv: https://arxiv.org/abs/1901.04827v2.

4.2 Finite-dimensional approximation of Gaussian processes with noisy observations

4.2.1 Finite-dimensional approximation in 1D

Here, we aim at imposing inequality constraints on GPs when observations are considered noisy. As an example, Figure 4.1 shows three types of GPs Y with training points at $x_1 = 0.2$, $x_2 = 0.5$, $x_3 = 0.8$, and different inequality conditions. We use a SE kernel with covariance parameters $\boldsymbol{\theta} = (\sigma^2 = 0.5^2, \ell = 0.2)$. We set a noise variance to be equal to 0.5% of the variance parameters σ^2. One can observe that different types of (constrained) Gaussian priors (top) yield different GP emulators (bottom) for the same training data. Note that the interpolation constraints are relaxed due to the noise effect, and that the inequality constraints are still satisfied everywhere.

Let $\{Y(x); x \in \mathcal{D}\}$ be a centred GP with arbitrary covariance function k and compact space $\mathcal{D} = [0, 1]$. As in Chapter 3, we consider a spline decomposition with an equispaced set of knots $t_1, \cdots, t_m \in \mathcal{D}$ such that $t_j = (j-1)/(m-1)$ for $j = 1, \cdots, m$. In contrast to Chapter 3, here we consider noisy observations $y_i \in \mathbb{R}$ for $i = 1, \cdots, n$.

Define Y_m as a the finite-dimensional GP consisting of the piecewise-linear interpolation of Y at knots $(t_1, \cdots, t_m)$:

$$Y_m(x) = \sum_{j=1}^{m} \xi_j \phi_j(x), \quad \text{s.t.} \quad Y_m(x_i) + \varepsilon_i = y_i \text{ (interpolation constraints)}, \tag{4.1}$$

where $x_i \in \mathcal{D}$, $\varepsilon_i \sim \mathcal{N}(0, \tau^2)$ for $i = 1, \cdots, n$, with noise variance τ^2, $\xi_j = Y(t_j)$ for $j = 1, \cdots, m$, and $\phi_1, \cdots, \phi_m$ are hat basis functions given by (3.2). As in many classical

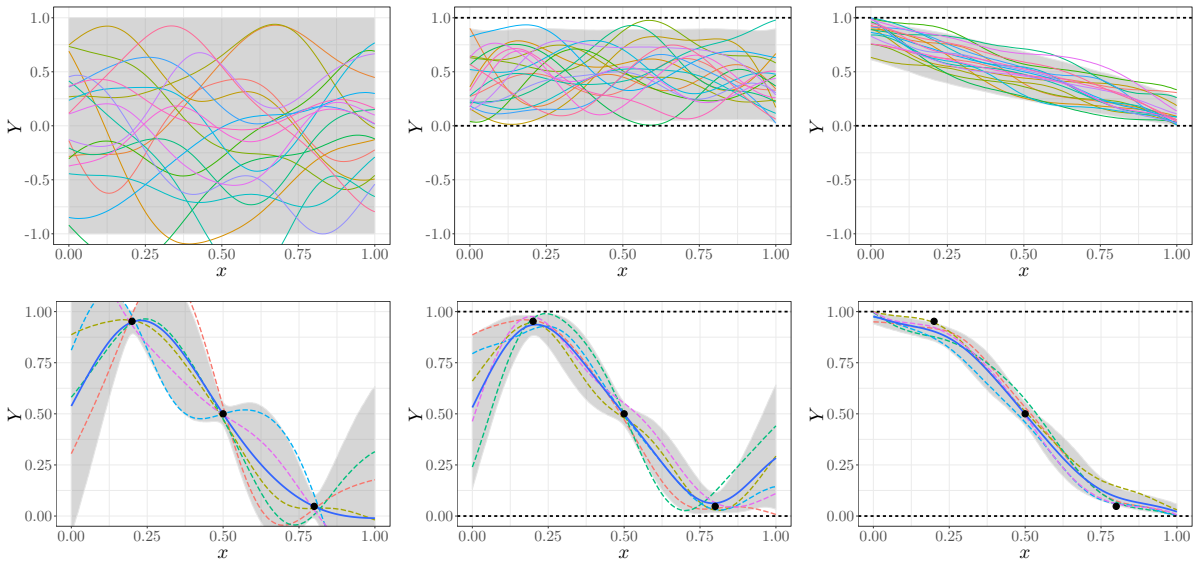


Figure 4.1: GP models under no constraints (left), boundedness constraints $Y \in [0, 1]$ (centre), boundedness and non-increasing constraints (right). Samples from the different types of (constrained) Gaussian priors and resulting GP emulators are shown in the first and second row, respectively. Each panel shows: the conditional emulations (dashed lines), and the 95% prediction interval (grey region). For boundedness constraints, bounds at $l = 0$ and $u = 1$ correspond to horizontal dashed lines. For the GP emulators, the conditional mean (blue solid line) and interpolation points (dots) are shown.

GP implementations (Rasmussen and Williams, 2005), we assume that $\varepsilon_1, \cdots, \varepsilon_n$ are independent, and independent of Y. However, since the framework proposed here does not have any restriction on the type of the covariance function, the extension to other noise distributions and/or noise with autocorrelation can be done as in standard GP implementations (Murphy, 2012; Rasmussen and Williams, 2005).

Note that the benefit of considering noisy observations in (4.1) is that, due to the "relaxation" of the interpolation conditions, the number of knots m does not have to be larger than the number of interpolation points n (assumption required in Chapter 3 for the interpolation of noise-free observations). Then, for $m \ll n$, the finite representation in (4.1) would lead to less expensive procedures since the cost of the MC and MCMC samplers grow with the value of m rather than n (see subsection 4.2.2).

4.2.2 Conditioning to linear inequality constraints

Now, assume that Y_m also satisfies inequality constraints everywhere in the input space (e.g. boundedness, monotonicity, convexity), i.e. $Y_m \in \mathcal{E}$ where $\mathcal{E}$ is a convex set of functions defined by some inequality conditions as in (3.3). As discussed in Chapter 3, the benefit of using (4.1) is that, for many constraint sets $\mathcal{E}$, satisfying $Y_m \in \mathcal{E}$ is equivalent to satisfying only a finite number of constraints at the knots, i.e. $Y_m \in \mathcal{E} \Leftrightarrow \boldsymbol{\xi} \in \mathcal{C}$. Here,

we consider the case where $\mathcal{C}$ is composed by a set of q linear inequalities given by (3.10).

We now aim at computing the distribution of Y_m conditionally on both the observations and inequality constraints. Let the vector $\boldsymbol{\xi}$ be a centred Gaussian vector with covariance matrix $\boldsymbol{\Gamma} = (k(t_i, t_j))_{1\leq i,j\leq m}$. Denote $\boldsymbol{\Phi}$ the $n \times m$ matrix defined by $\boldsymbol{\Phi}_{i,j} = \phi_j(x_i)$, and $\boldsymbol{y} = [y_1, \cdots, y_n]^\top$ the vector of noisy observations at points $x_1, \cdots, x_n$. Then, the distribution of $\boldsymbol{\xi}$ conditioned on $\boldsymbol{\Phi\xi} + \boldsymbol{\varepsilon} = \boldsymbol{y}$, with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$, is given by[1]

$$\boldsymbol{\xi}|\{\boldsymbol{\Phi\xi} + \boldsymbol{\varepsilon} = \boldsymbol{y}\} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{4.2}$$

where

$$\boldsymbol{\mu} = \boldsymbol{\Gamma\Phi}^\top[\boldsymbol{\Phi\Gamma\Phi}^\top + \tau^2\mathbf{I}]^{-1}\boldsymbol{y}, \quad \text{and} \quad \boldsymbol{\Sigma} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma\Phi}^\top[\boldsymbol{\Phi\Gamma\Phi}^\top + \tau^2\mathbf{I}]^{-1}\boldsymbol{\Phi\Gamma}. \tag{4.3}$$

One can note that, in the limit as the noise variance $\tau^2 \to \infty$, then $\boldsymbol{\mu} \to \mathbf{0}$ and $\boldsymbol{\Sigma} \to \boldsymbol{\Gamma}$, and therefore the distribution in (4.2) ignores the observations $\boldsymbol{y}$. In that case, MC and MCMC samplers are performed in the sample space of the prior of $\boldsymbol{\xi}$, which is less restrictive than the one of $\boldsymbol{\xi}|\{\boldsymbol{\Phi\xi} + \boldsymbol{\varepsilon} = \boldsymbol{y}\}$.

As in Chapter 3, denote $\boldsymbol{\Lambda} = (\lambda_{k,j})_{1\leq k\leq q, 1\leq j\leq m}$, $\boldsymbol{l} = (\ell_k)_{1\leq k\leq q}$, and $\boldsymbol{u} = (u_k)_{1\leq k\leq q}$. Since the inequality constraints are on $\boldsymbol{\Lambda\xi}$, one can first show that the posterior distribution of $\boldsymbol{\Lambda\xi}$ conditioned on $\boldsymbol{\Phi\xi} + \boldsymbol{\varepsilon} = \boldsymbol{y}$ and $\boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u}$ is truncated Gaussian-distributed (see Chapter 3 for further discussion when noise-free observations are considered), i.e.

$$\boldsymbol{\Lambda\xi}|\{\boldsymbol{\Phi\xi} + \boldsymbol{\varepsilon} = \boldsymbol{y}, \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u}\} \sim \mathcal{TN}\left(\boldsymbol{\Lambda\mu},\ \boldsymbol{\Lambda\Sigma\Lambda}^\top,\ \boldsymbol{l},\ \boldsymbol{u}\right). \tag{4.4}$$

Notice that the inequalities are encoded in the posterior mean $\boldsymbol{\Lambda\mu}$, the posterior covariance $\boldsymbol{\Lambda\Sigma\Lambda}^\top$, and the bounds $(\boldsymbol{l}, \boldsymbol{u})$. Moreover, one must highlight that by considering noisy observations, due to the "relaxation" of the interpolation conditions, inequality constraints can be imposed also when the observations $y_1, \cdots, y_n$ do not fulfil the inequalities. Note that the mode of (4.4) is given by (3.15) but with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ as in (4.3). Finally, Algorithm 1 can be applied using (4.3) and (4.4).

Remark. The extension to d dimensions is given by tensorisation as in subsection 3.2.4. We only need to write the interpolation conditions of the form $Y_{m_1,\cdots,m_d}(\boldsymbol{x}_i) + \varepsilon_i = y_i$ with $\boldsymbol{x}_i \in \mathcal{D}$, $y_i \in \mathbb{R}$ and $\varepsilon_i \sim \mathcal{N}(0, \tau^2)$ for $i = 1, \cdots, n$. Then, Algorithm 1 can be used. Notice that having less knots than observations can have a great impact since the MC and MCMC samplers will then be performed in low dimensional spaces when $m = m_1 \times \cdots \times m_d \ll n$. Furthermore, in that case, the inversion of the matrix $(\boldsymbol{\Phi\Gamma\Phi}^\top + \tau^2\mathbf{I}) \in \mathbb{R}^{n\times n}$ can be computed more efficiently through the matrix inversion lemma (Press et al., 1992), reducing the computational complexity to the inversion of an $m \times m$ full-rank matrix. Therefore, the computation of the conditional distribution in (4.2) and the estimation of the covariance parameter can be speeded up. Moreover, due to the relaxation of the interpolation conditions through a noise effect, MC and MCMC samplers are performed in less restrictive sample spaces, and this leads to faster emulators.

[1] Further developments can be generalised for a vector of noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Omega})$ with noise covariance matrix $\boldsymbol{\Omega} = (k_{\text{noise}}(x_i, x_j))_{1\leq i,j\leq n}$. In that case, we only need to replace the noise term $\tau^2\boldsymbol{I}$ by $\boldsymbol{\Omega}$.

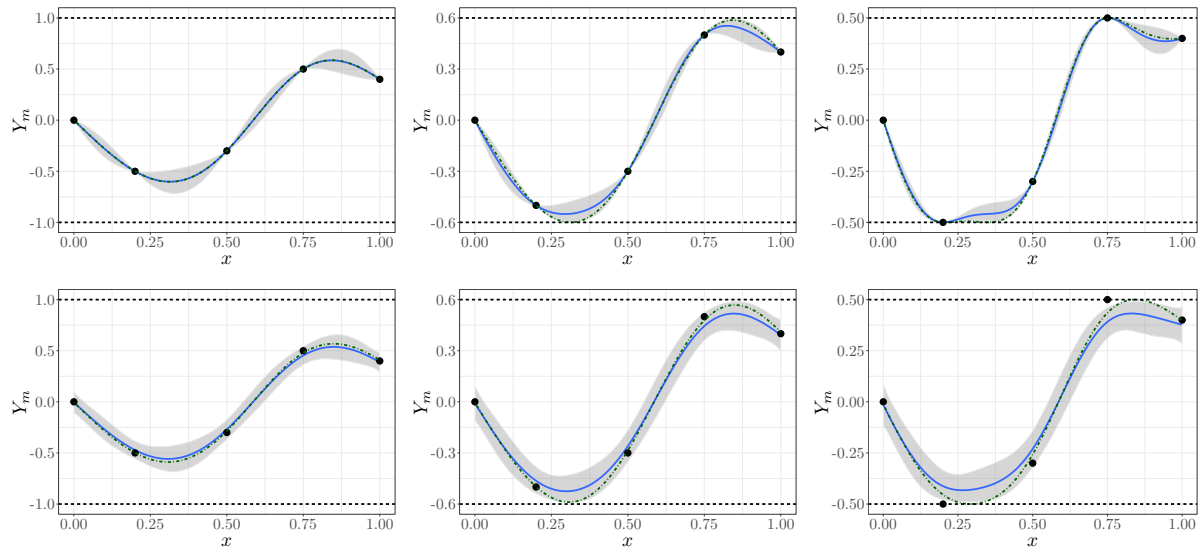


Figure 4.2: GP emulators under boundedness constraints. Results are shown considering (top) noise-free and (bottom) noisy observations: (left) $Y_m \in [-1, 1]$, (centre) $Y_m \in [-0.6, 0.6]$, and (right) $Y_m \in [-0.5, 0.5]$. Each panel shows: the observations (dots), the conditional mean (blue solid line), the conditional mode (green dot-dash line), the 95% prediction interval (grey region), and the bounds (dashed lines).

4.2.3 Numerical illustrations

4.2.3.1 1D toy example under boundedness constraints

Here, we use the GP framework introduced in Section 4.2 for emulating bounded trajectories $Y_m \in [-\alpha, \alpha]$ with constant $\alpha \in \mathbb{R}^+$. We aim at analysing the constrained GP emulator when noise-free or noisy observations are considered. The dataset is $(x_i, y_i)_{1 \leq i \leq 5}$: $(0, 0)$, $(0.2, -0.5)$, $(0.5, -0.3)$, $(0.75, 0.5)$, and $(1, 0.4)$. We use a Matérn 5/2 covariance function with fixed variance parameter $\sigma^2 = 10$, leading to highly variable trajectories. The length-scale parameter ℓ and the noise variance τ^2 are estimated via ML.

The effect of different bounds $[-\alpha, \alpha]$ on the constrained GPs can be seen in Figure 4.2. We set $m = 100$ for having emulations with high-quality of resolution, and we generate 10^4 constrained emulations via RSM (Maatouk and Bay, 2016). Observe that, since interpolation conditions are relaxed due to the influence of the noise variance τ^2, the prediction intervals are wider when bounds become closer to the observations. When $\alpha = 0.5$, the noise-free GP emulator yields costly procedures due to a small acceptance rate equal to 0.1%. In contrast, when noisy observations are assumed, emulations are more likely to be accepted leading to an acceptance rate equal to 16.92%.

Now, we assess both MC and MCMC methods from subsection 3.3.2 for the approximation of the truncated Gaussian posterior distribution in (4.4). We consider the examples in Figure 4.2. For the MCMC samplers, we use the posterior mode solution from (3.14) as the starting state of the Markov chains. This initialises the chains in a high

probability region. Therefore, only few emulations have been "burned" in order to have samples that appeared to be independent of the starting state. Here, we burn the first 100 emulations. As in subsection 3.3.2, we evaluate the performance of both MC and MCMC samplers in terms of the ESS. One must remember that the ESS indicator gives an intuition on how many emulations of the sample path can be considered independent (Gong and Flegal, 2016). We compute the ESS of each coordinate of $\boldsymbol{\xi} \in \mathbb{R}^m$, i.e. $\text{ESS}_j = \text{ESS}(\xi_j^1, \cdots, \xi_j^{n_s})$ for $j = 1, \cdots, m$, and we evaluate the quantiles $(q_{10\%}, q_{50\%}, q_{90\%})$ over the m resulting ESS values. The sample size $n_s = 10^4$ has been chosen to be larger than the minimum ESS required to obtain a proper estimation of the vector $\boldsymbol{\xi} \in \mathbb{R}^m$ (Gong and Flegal, 2016). Finally, we test the efficiency of each sampler by computing the TN-ESS at $q_{10\%}$: TN-ESS $= q_{10\%}(\text{ESS})/(\text{CPU Time})$.

Table 4.1 displays the performance indicators obtained for each MC and MCMC samplers. Firstly, one can observe that RSM yielded the most expensive procedures due to its high rejection rate when sampling the constrained trajectories from the posterior mode. In particular, for $\alpha = 0.5$, and assuming noise-free observations, the prohibitively small acceptance rate of RSM led to costly procedures (about 7 hours) making it impractical. Secondly, although the Gibbs sampler needs to discard intermediate samples (thinning effect), it provided accurate ESS values within a moderate running time (with effective sampling rates of 400 s^{-1}). Thirdly, due to the high acceptance rates obtained by ExpT, and good exploratory behaviour of the exact HMC, both samplers provided much more efficient TN-ESS values compared to their competitors, generating thousands of effective emulations per second. Finally, as we expected, the performance of some samplers were improved when adding noise. For RSM, due to the relaxation of the interpolation conditions, we noted that emulations were more likely to be accepted leading to quicker routines: more than 150 times faster with noise (see Table 4.1, for $\alpha = 0.5$).

Finally, we assess the efficiency of the HMC sampler in terms of its mixing performance (see Figure 4.3). We analyse the example in Figure 4.2 using the noisy GP emulator with $\alpha = 0.5$. From both the trace and autocorrelation plots at $Y_m(0.01)$, one can conclude that the HMC sampler mixes well with small correlations.

4.2.3.2 1D Toy example under multiple constraints

In Chapter 3, numerical implementations were limited to noise-free observations that fulfilled the inequality constraints. Here, we test the case when noisy observations do not necessarily satisfy the inequalities.

Consider the sigmoid function given by

$$x \mapsto \frac{1}{1 + \exp\{-10(x - \frac{1}{2})\}}, \quad \text{for} \quad x \in [0, 1]. \tag{4.5}$$

We evaluate (4.5) at $n = 300$ random values of x, and we contaminate the function evaluations with an additive Gaussian white noise with a standard deviation equal to 10% of the sigmoid range. Since (4.5) exhibits both boundedness and non-decreasing

Table 4.1: Efficiency of MC and MCMC for emulating bounded samples $Y_m \in [-\alpha, \alpha]$. Best results are shown in bold. For the Gibbs sampler, we set the thinning parameter to 200 emulations in order to obtain competitive ESS values w.r.t. other samplers. †Results could not be obtained due to numerical instabilities.

Bounds	Method	Without noise variance			With noise variance		
		CPU Time $[s]$	ESS $[\times 10^4]$ $(q_{10\%}, q_{50\%}, q_{90\%})$	TN-ESS $[\times 10^4 s^{-1}]$	CPU Time $[s]$	ESS $[\times 10^4]$ $(q_{10\%}, q_{50\%}, q_{90\%})$	TN-ESS $[\times 10^4 s^{-1}]$
$\alpha = 1$	RSM	61.30	(0.97, 1.00, 1.00)	0.02	57.64	(0.91, 1.00, 1.00)	0.02
	ExpT	2.30	(0.98, 1.00, 1.00)	0.43	2.83	(0.96, 1.00, 1.00)	0.34
	Gibbs	19.70	(0.84, 0.86, 0.91)	0.04	21.18	(0.75, 0.84, 0.91)	0.04
	HMC	**1.89**	(0.95, 0.99, 1.00)	**0.50**	**1.92**	(0.94, 0.99, 1.00)	**0.49**
$\alpha = 0.75$	RSM	63.59	(1.00, 1.00, 1.00)	0.02	48.66	(0.95, 0.99, 1.00)	0.02
	ExpT	3.22	(0.96, 0.99, 1.00)	0.30	3.24	(0.98, 1.00, 1.00)	0.30
	Gibbs	20.20	(0.83, 0.86, 0.91)	0.04	18.23	(0.74, 0.84, 0.93)	0.04
	HMC	**1.46**	(0.94, 1.00, 1.00)	**0.64**	**1.28**	(0.94, 0.97, 1.00)	**0.73**
$\alpha = 0.6$	RSM	242.34	(0.94, 0.97, 1.00)	0	101.20	(0.96, 1.00, 1.00)	0.01
	ExpT	2.94	(0.94, 1.00, 1.00)	0.32	2.80	(0.98, 1.00, 1.00)	0.35
	Gibbs	18.89	(0.80, 0.83, 0.94)	0.04	18.90	(0.77, 0.84, 0.92)	0.04
	HMC	**1.72**	(0.92, 0.99, 1.00)	**0.53**	**1.68**	(0.93, 0.96, 1.00)	**0.55**
$\alpha = 0.5$	RSM	25512.77	(0.98, 1.00, 1.00)	0	157.06	(0.96, 0.99, 1.00)	0.01
	ExpT	**2.50**	(0.99, 1.00, 1.00)	**0.40**	**2.69**	(0.97, 1.00, 1.00)	**0.36**
	Gibbs†	—	—	—	—	—	—
	HMC	6.20	(0.86, 0.90, 0.98)	0.14	2.14	(0.52, 0.85, 0.97)	0.24

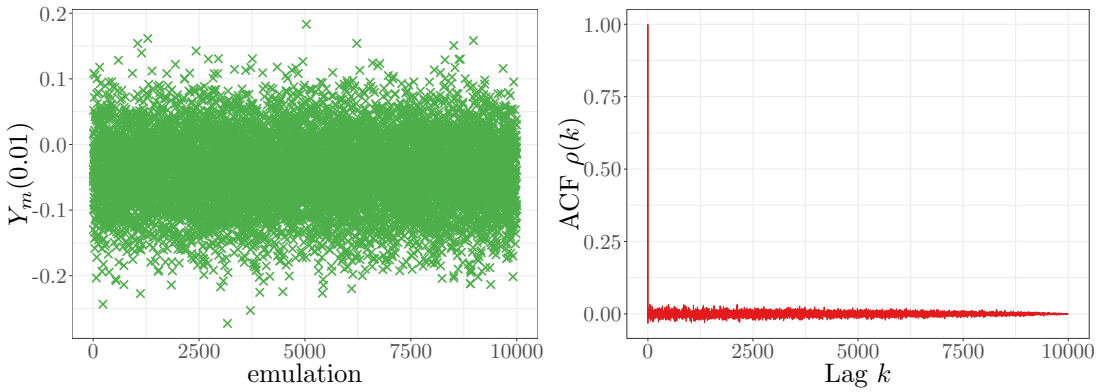


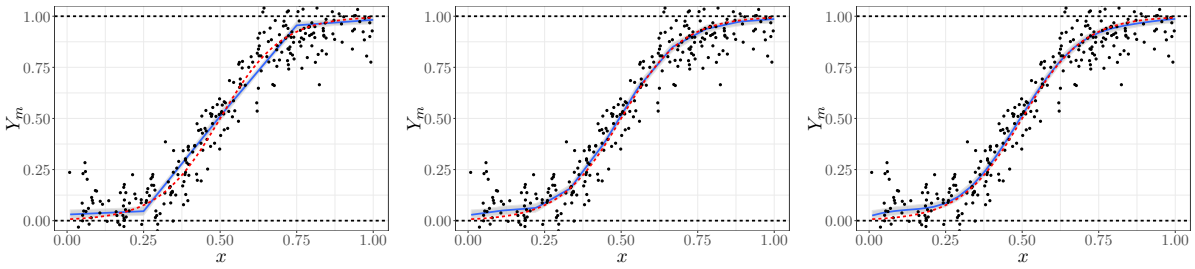
Figure 4.3: Efficiency of the HMC sampler in terms of its mixing performance. Results are shown for the (left) trace and (right) autocorrelation plots at $Y_m(0.01)$.

conditions, we add those constraints into the GP emulator Y_m using the convex set:

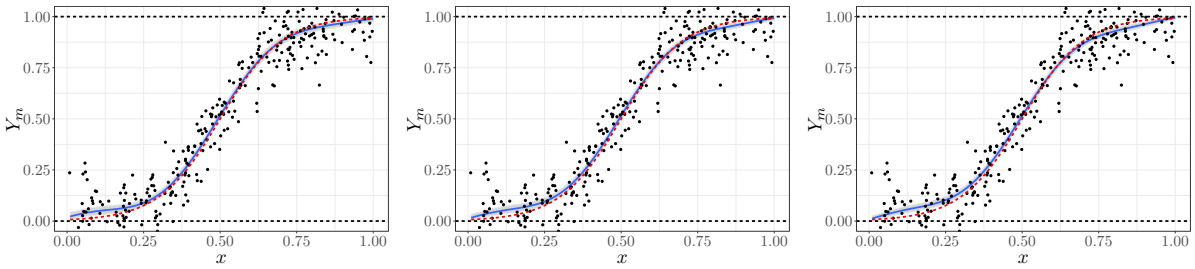
$$\mathcal{C}^{\uparrow}_{[0,1]} = \left\{ \mathbf{c} \in \mathbb{R}^m;\ \forall j = 2, \cdots, m\ :\ c_j \geq c_{j-1},\ c_1 \geq 0,\ c_m \leq 1 \right\}.$$

Hence, MC/MCMC samplers are performed on $\mathbb{R}^{m+1}$. As a covariance function, we use a SE kernel with parameters (σ^2, ℓ, τ^2) estimated via ML.

Unlike Chapter 3, there is no need here to satisfy the condition $m \geq n$, due to the noise. Therefore, the finite approximation in Section 4.2 can be seen as a surrogate



(a) $m = 5$, CPU Time $= 0.10$ s (b) $m = 10$, CPU Time $= 0.16$ s (c) $m = 25$, CPU Time $= 0.21$ s



(d) $m = 50$, CPU Time $= 0.39$ s (e) $m = 100$, CPU Time $= 1.37$ s (f) $m = 150$, CPU Time $= 4.40$ s

Figure 4.4: GP emulators under boundedness and monotonicity constraints. Results are shown for different amounts of knots m. Each panel shows: the target function (dashed lines), the noisy training points (dots), the conditional mean (solid line), the 95% prediction interval (grey region), and the bounds (horizontal dashed lines).

model of standard GP emulators for $m \ll n$. Figure 4.4 shows the performance of the constrained emulators via HMC for $m = 5, 10, 25, 50, 100, 150$. For smaller values of m, the GP emulator runs fast but with a low quality of resolution of the approximation. For example, for $m = 5$ and $m = 10$, because of the linearity assumption between knots, the predictive mean presents breakpoints at the knots. On the other hand, the GP emulator yields smoother (constrained) emulations as m increases ($m \geq 25$). In particular, one can observe that for $m = 25$, the emulator leads to a good trade-off between quality of resolution and running time (6.5 times faster than for $m = 100$).

Finally, we test the performance of the proposed framework under different regularity assumptions, noise levels and inequality constraints. For the example in Figure 4.4, we fixed $m = 200$ and used different choices of covariance functions. Given a fixed noise level, the covariance parameters of each GP model, i.e. $\boldsymbol{\theta} = (\sigma^2, \ell)$, were estimated via ML. The noise levels were chosen using different proportions of the sigmoid range. We assessed the proposed GP emulator accounting for either boundedness constraints, monotonicity constraints or both. We computed the CPU time and the Q^2 criterion in (3.17). We used the 300 noise-free function evaluations from (4.5) as test data. Results are shown in Table 4.2. One can note that the introduction of noise let us also have constrained GP emulations in the cases where the regularity of the GP prior is not in agreement with the regularity of data and the inequality conditions. In particular, expensive procedures

Table 4.2: Performance of the GP emulators from Figure 4.4 under different regularity assumptions, noise levels and inequality constraints. The noise levels are chosen using different proportions of the range of the sigmoid function in (4.5). CPU Time [s] and Q^2 [%] results are shown for various covariance function (i.e. Matérn 3/2 kernel, Matérn 5/2 kernel and SE kernel), and different inequality constraints.

Noise level	Boundedness Constraints					
	Matérn $\frac{3}{2}$		Matérn $\frac{5}{2}$		SE	
	Time	Q^2	Time	Q^2	Time	Q^2
0%	—	—	—	—	—	—
0.5%	1.0	99.4	0.8	99.6	0.6	99.7
1.0%	1.1	99.4	0.7	99.6	0.6	99.7
5.0%	1.0	98.9	0.8	99.3	0.6	99.5
10.0%	0.9	98.2	0.8	98.9	0.6	99.2

Noise level	Monotonicity Constraints					
	Matérn $\frac{3}{2}$		Matérn $\frac{5}{2}$		SE	
	Time	Q^2	Time	Q^2	Time	Q^2
0%	—	—	—	—	—	—
0.5%	117.0	99.5	1.4	99.8	1.2	99.8
1.0%	14.5	99.1	1.2	99.8	1.0	99.8
5.0%	7.4	95.6	1.0	99.3	0.8	99.3
10.0%	6.3	91.9	1.0	98.7	0.6	98.9

Noise level	Boundedness & Monotonicity Constraints					
	Matérn $\frac{3}{2}$		Matérn $\frac{5}{2}$		SE	
	Time	Q^2	Time	Q^2	Time	Q^2
0%	—	—	—	—	—	—
0.5%	—	—	17.3	99.7	13.9	99.8
1.0%	$> 10^4$	99.4	15.2	99.6	10.4	99.6
5.0%	251.8	96.7	13.3	98.6	8.6	98.3
10.0%	246.1	94.6	13.3	97.5	8.6	97.0

were obtained for the Matérn 3/2 kernel when considering monotonicity. In those cases, the high irregularity of the (unconstrained) GP prior yielded more restrictive sample spaces that fulfil the monotonicity conditions. Furthermore, one may observe that the computational cost of emulators can be attenuated by increasing the noise level but at the cost of the accuracy of predictions.

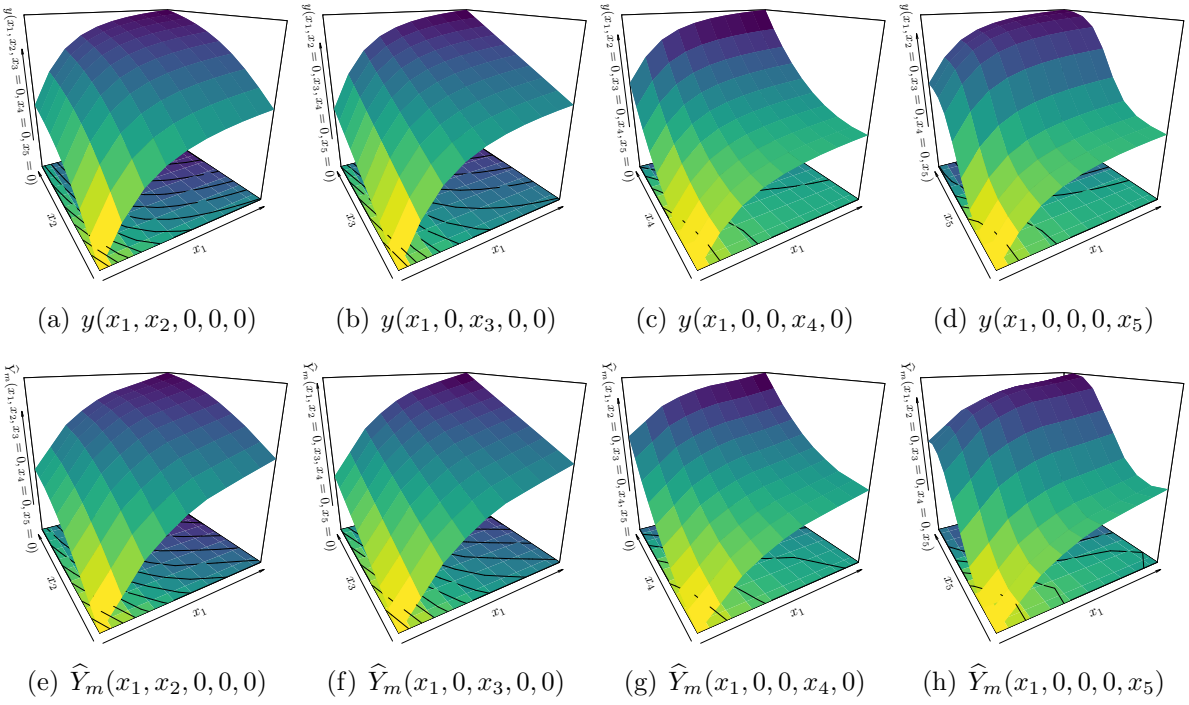
4.2.3.3 Illustration on a 5D monotonic example

In Chapter 3, since the approximation of GPs was introduced to interpolate a given number of (noise-free) observations, it was strictly necessary to have more knots than observations, i.e. $m \geq n$. Moreover, the interpolation conditions yielded more restricted domains where samplers could not be performed efficiently. Therefore, the extension to high dimensions was challenging there (limited to $d = 2$). Here, we show that, by considering noisy observations, constrained GP models can be performed in higher dimensions and for a high number of observations within a reasonable running time.

Consider the 5D target function given by

$$y(\boldsymbol{x}) = \arctan(5x_1) + \arctan(2x_2) + x_3 + 2x_4^2 + \frac{2}{1 + \exp\{-10(x_5 - \frac{1}{2})\}}, \tag{4.6}$$

with $\boldsymbol{x} \in [0,1]^5$. One can observe that y is non-decreasing w.r.t. all its input variables x_i for $i = 1, \cdots, 5$. Although some computations could be potentially simplified given that y is additive, we do not take advantage of that aiming at testing the framework on



(a) $y(x_1, x_2, 0, 0, 0)$ (b) $y(x_1, 0, x_3, 0, 0)$ (c) $y(x_1, 0, 0, x_4, 0)$ (d) $y(x_1, 0, 0, 0, x_5)$

(e) $\widehat{Y}_m(x_1, x_2, 0, 0, 0)$ (f) $\widehat{Y}_m(x_1, 0, x_3, 0, 0)$ (g) $\widehat{Y}_m(x_1, 0, 0, x_4, 0)$ (h) $\widehat{Y}_m(x_1, 0, 0, 0, x_5)$

Figure 4.5: GP emulators under monotonicity constraints in 5D. True and predictive mean profiles are shown in the first and second row, respectively.

a fully 5D example. We evaluate y on a maximin Latin-hypercube DoE over $[0, 1]^5$ at 2000 locations using (Dupuy et al., 2015).

As in subsection 4.2.3, we contaminate the resulting observations with an additive Gaussian white noise with a standard deviation equal to 1% of the range of y. As covariance function, we use a 5D SE kernel and estimate the parameters $(\boldsymbol{\theta}, \tau^2)$ via ML.[2] We set the number of knots as a trade-off between high-quality of resolution in the approximation and CPU running time. Thus, we fix the number of knots per dimension to five, except for the third and fifth dimensions, i.e. $m_\kappa = 5$ for $\kappa = 1, 2, 4$. For the third dimension, we let $m_3 = 2$ due to the linearity of the target function y w.r.t. x_3. For the fifth dimension, we fix $m_5 = 7$ since y varies more through this dimension compared to the other ones. This leads to a total number of 1750 knots.

Figure 4.5 shows the predictions obtained by the constrained GP emulator under monotonicity conditions. The proposed emulator is obtained via HMC, leading to a CPU running time of 44.8 minutes for generating 10^4 monotonic trajectories. The ESS values are equal to $(92.9\%, 98.1\%, 100\%)$ at quantiles $(q_{10\%}, q_{50\%}, q_{90\%})$, providing an effective sample rate of 3.5 s^{-1} at quantile $q_{10\%}$. Besides the efficient and fast performance of HMC, one can observe from Figure 4.5 that the GP emulator was able to capture the non-decreasing dynamics of the target function everywhere.

[2] 5D SE kernel: $k_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\Big\{ - \sum_{i=1}^{5} \frac{(x_i - x_i')^2}{2\ell_i^2} \Big\}$ with $\boldsymbol{\theta} = (\sigma^2, \ell_1, \cdots, \ell_5)$.

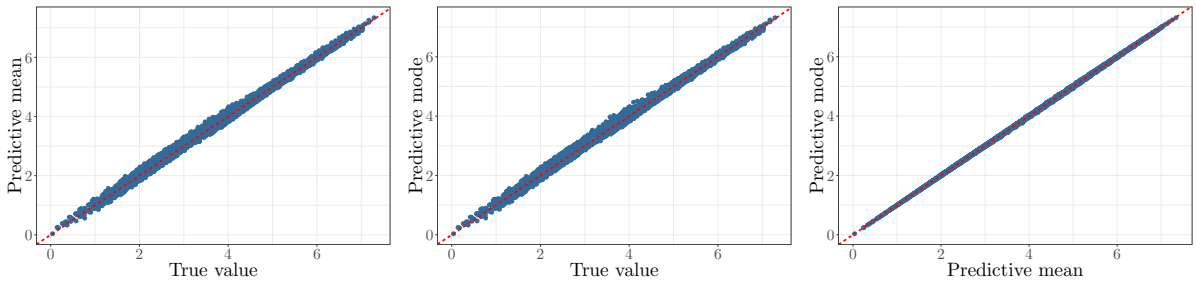


Figure 4.6: Quality of predictions from Figure 4.5: predictive mean vs true observations (left), predictive mode vs true observations (centre), and predictive mode vs predictive mean (right).

Finally, we evaluate the quality of predictions using the Q^2 criterion. We test both the predictive mean and mode, obtaining Q^2 values equal to 99.57% and 99.56% (respectively). Figure 4.6 plots the predictive mean and the predictive mode vs the true (noise-free) observations. Note that, since the predictive mode of (3.14) can be obtained much faster than the predictive mean (requiring only a couple of seconds), one may suggest it as an accurate point prediction of the conditional process.

4.3 Coastal flooding applications

Coastal flooding models based on GP emulators have taken great attention regarding computational simplifications for estimating flooding indicators (like the maximum water level at the coast, discharge, flood spatial extend, etc.) (see, e.g., Azzimonti et al., 2019; Rohmer and Idier, 2012). However, since standard GP emulators do not take into account the nature of many coastal flooding events satisfying positivity and/or monotonicity constraints, those approaches often require a large number of observations (commonly costly to obtain) in order to obtain reliable predictions. In those cases, GP emulators yield expensive procedures. Here we show that, by enforcing GP emulators to those inequality constraints, the GP framework can lead to more reliable prediction also when a small amount of data is available.

We test the performance of the emulator in Section 4.2 on two coastal flooding datasets provided by the BRGM (which is the French Geological Survey, "Bureau de Recherches Géologiques et Minières", in French). The first dataset corresponds to a 2D coastal flooding application located on the Mediterranean coast, focusing on the water level at the coast (Rohmer and Idier, 2012). The second one describes a 5D coastal flooding example induced by overflow on the Atlantic coast, focusing on the inland flooded surface (Azzimonti et al., 2019). We trained different GP emulators whether the inequality constraints are considered or not. For the unconstrained emulators, we use the GP-based scheme provided by the R package `DiceKriging` (Roustant et al., 2012).

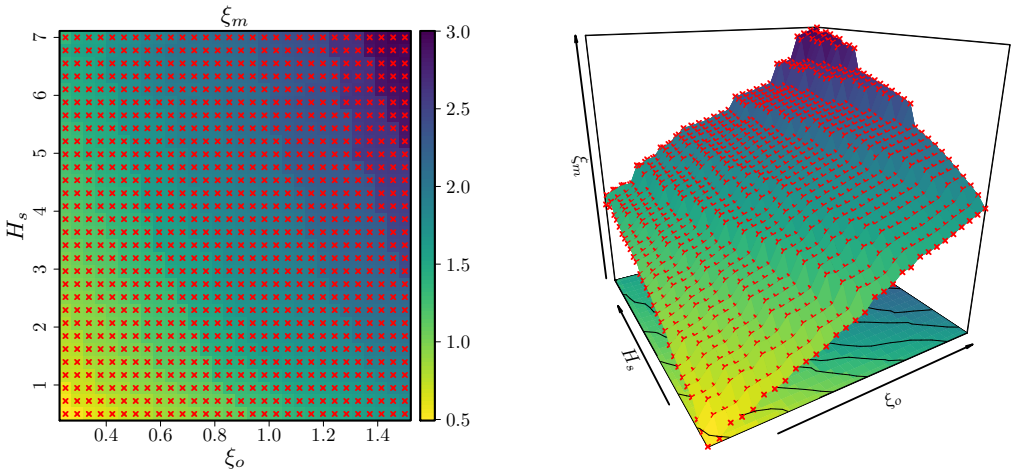


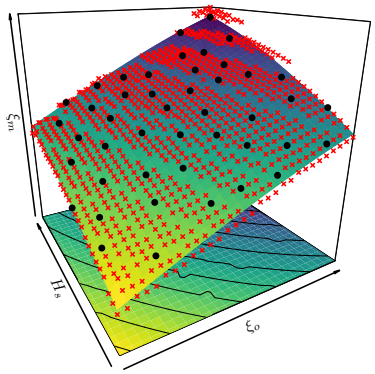
Figure 4.7: 2D coastal flooding application. (Left) 2D visualisation of the ξ_m values measured over a regular grid. (Right) 3D visualisation of the ξ_m data.

4.3.1 2D coastal flooding on the Mediterranean coast

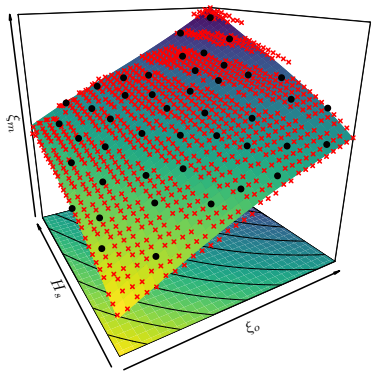
The coastal study site is located on a lido, which has faced two flood events in the past (Rohmer and Idier, 2012). The dataset used here contains 900 observations of the maximum water level at the coast ξ_m depending on two input parameters: the offshore water level (ξ_o) and the wave height (H_s), both in metre units. The observations are taken within the domains $\xi_o \in [0.25, 1.50]$ and $H_s \in [0.5, 7]$ (with each dimension being discretised in 30 elements). One must note that, on the domain considered for the input variables, ξ_m increases as ξ_o and H_s increase (see Figure 4.7).

Here, we normalised the input space to be in $[0, 1]^2$. As covariance function, we used 2D SE kernels with parameters $\boldsymbol{\theta} = (\sigma^2, \ell_1, \ell_2)$. Both $\boldsymbol{\theta}$ and the noise variance τ^2 were estimated via ML. For the constrained model, we proposed emulators accounting for both positivity and monotonicity, and we manually fixed the number of knots $m_1 = m_2 = 25$.

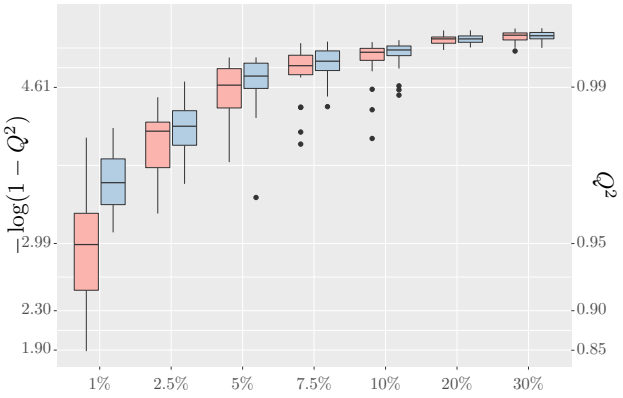
For illustrative purposes, we first trained both unconstrained and constrained GP emulators using 5% of the data (equivalent to 45 training points chosen by a maximin Latin hypercube DoE), and we aimed at predicting the remaining 95%. Results are in Figures 4.8(a) and 4.8(b). In particular, one can observe that the constrained GP emulator slightly outperformed the prediction around the extreme values of ξ_m, leading to an absolute improvement of 4% of the Q^2 indicator. Then, we repeated the experiment using twenty different sets of training data and different proportions of training sets. According to Figure 4.8(c), one can observe that the constrained emulator often outperforms the unconstrained one, with significant Q^2 improvements for small training sets. As coastal flooding simulators are commonly costly-to-evaluate, the benefit of having accurate prediction with lesser number of observations becomes useful for practical implementations.



(a) Unconstrained GP: $Q^2 = 0.987$



(b) Constrained GP: $Q^2 = 0.991$



(c) Q^2 performance

Figure 4.8: 2D GP emulators for modelling the coastal flooding data in (Rohmer and Idier, 2012). (Left) Prediction results using 5% of the dataset via maximin Latin hypercube DoE. Each panel shows: training and test points (black dots and red crosses), the conditional mean function (solid surface), and the Q^2 criterion (subcaptions). (c) Q^2 assessment using different proportions of training points (x-axis) and using twenty different random training sets. Results are shown for the unconstrained (red) and constrained (blue) GP emulators.

4.3.2 5D coastal flooding on the Atlantic coast

As in (Azzimonti et al., 2019), here we focus on the coastal flooding induced by overflow. We consider the "Boucholeurs" area located close to "La Rochelle", France. This area was flooded during the 2010 Xynthia storm, an event characterised by a high storm surge in phase with a high spring tide. We focus on those primary drivers, and on how they affect the resulting flooded surface.

The dataset contains 200 observations of the flooded area Y in m^2 depending on five input parameters $\mathbf{x} = (T, S, \phi, t_+, t_-)$ detailing the offshore forcing conditions:

- The tide is simplified by a sinusoidal signal parametrised by its high tide level $T \in [0.95, 3.70]$ (m).
- The surge signal is described by a triangular model using four parameters: the peak amplitude $S \in [0.65, 2.50]$ (m), the phase difference $\phi \in [-6, 6]$ (hours), between the surge peak and the high tide, the time duration of the raising part $t_- \in [-12.0, -0.5]$ (hours), and the falling part $t_+ \in [0.5, 12.0]$ (hours).

The dataset is freely available in the R package `profExtrema` (Azzimonti, 2018). One must note that the flooded area Y increases as T and S increase.

Before implementing the corresponding GP emulators, we first analysed the structure of the dataset. We tested various standard linear regression models in order to understand the influence of each input variable $\mathbf{x} = (T, S, \phi, t_+, t_-)$. We assessed the quality of the linear models using the adjusted R^2 criterion. Similarly to the Q^2 criterion, the R^2 indicator evaluates the quality of predictions over all the observation points rather than only over the training data. Therefore, for noise-free observations, the R^2 indicator is equal to one if predictors are exactly equal to the data. We also tested various models considering different input variables (e.g. transformation of variables, or inclusion of interaction terms). After testing different linear models, we observed that they were more sensitive to the inputs T and S rather than to other ones. We also noted that, by transforming the phase coordinate $\phi \mapsto \cos(2\pi\phi)$, an absolute improvement about 26% of the R^2 indicator was obtained, and the influence of both t_- and t_+ became more significant. Finally, we used these settings for the GP implementations.

We normalised the input space to be in $[0, 1]^5$, and we used a 5D Matérn 5/2 kernel. The covariance parameters $\boldsymbol{\theta} = (\sigma^2, \ell_1, \cdots, \ell_5)$ and the noise variance τ^2 were estimated via ML. We also tested other types of kernel structures, including SE and Matérn 3/2 kernels, but smaller Q^2 values were obtained. For the constrained model, we proposed GP emulators accounting for positivity constraints everywhere. We also imposed monotonicity constraints along the T and S input dimensions. Since the computational complexity of the constrained GP emulator increases with the number of knots m used in the piecewise-linear representation, we strategically fixed them in coordinates requiring high quality of resolution. Since we observed that the contribution of the inputs T, S, t_- and t_+ was almost linear (result in agreement with Azzimonti et al., 2019), we placed fewer number of knots over those entries. In particular, we fixed as number of knots per dimension: $m_1 = m_2 = 4$, $m_3 = 5$ and $m_4 = m_5 = 3$.

As in subsection 4.3.1, we trained GP emulators using twenty different sets of training data and different proportions of training data. According to Figure 4.9, one can observe once again that the constrained GP emulator often outperforms the unconstrained one, with significant Q^2 improvements for small training sets. In particular, one can note that, by enforcing the GP emulators with both positivity and monotonicity constraints, accurate predictions were also provided by using only 10% of the observations as training points (equivalent to 20 observations).

4.4 Conclusions

We have introduced a constrained GP emulator with linear inequality conditions and noisy observations. By relaxing the interpolation of observations through a noise effect, MC/MCMC samplers are performed in less restrictive sample spaces. This leads to faster emulators while preserving high effective sampling rates. As seen in the experiments, the HMC sampler from (Pakman and Paninski, 2014) usually outperformed its competitors, providing much more efficient effective sample rates in high dimensional sample spaces.

Since there is no need of having more knots than observations ($m \geq n$), the computa-

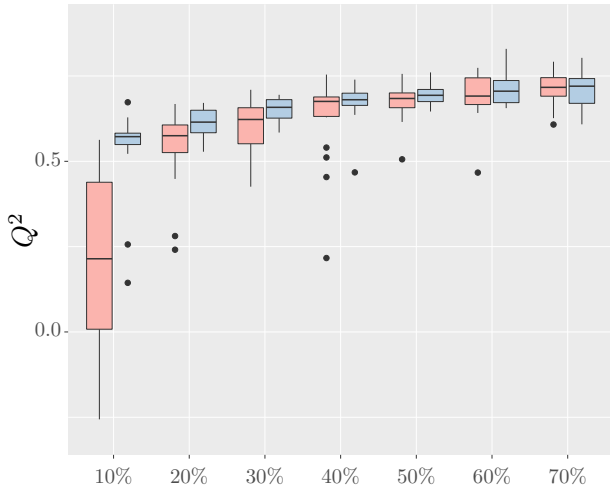


Figure 4.9: 5D GP emulators for modelling the coastal flooding data in (Azzimonti et al., 2019). The boxplots show the Q^2 results using different proportions of training points (x-axis) and using twenty different random training sets. Results are shown for the unconstrained (red) and constrained (blue) GP emulators.

tional complexity of MC and MCMC samplers is independent of n. Therefore, since the samplers are performed on $\mathbb{R}^m$, they can be used for large values of n by letting $m \ll n$. As shown in the 5D monotonic example, effective monotone emulations can be obtained within reasonable running times (about tens of minutes). On 2D and 5D coastal flooding applications, we also show that more flexible and realistic GP emulators can be obtained by considering noise effects and by enforcing the (linear) inequality constraints.

Despite the improvements obtained here for scaling GP emulators for $d > 2$, its tensor structure makes it impractical for tens of input variables. We believe that this limitation could be partially mitigated by considering supplementary assumptions on the nature of the target function in order to reduce the dimensionality of the sample spaces where MC/MCMC samplers are performed (e.g. additivity – see Chapter 6). In addition, other types of designs of the knots that scale better to high dimensions could be explored (e.g. triangular designs – see Chapter 10).

Part IV

Gaussian Processes under Inequality Constraints in High Dimensions

Chapter 5

Constrained Gaussian Processes using Free-Knot Designs

Contents

5.1 Introduction

Due to the tensor structure of the finite-dimensional GP in Chapter 3, it becomes costly as the number of knots m increases. Therefore, it is worth preferentially placing knots only in regions requiring a high quality of resolution (e.g., in highly variable regions). In that case, aiming a trade-off between quality of representation and computational complexity, one may expect that the model in (3.9) could be potentially performed in higher dimensions, e.g. when $d \geq 3$.

Here, we suggest the use of an alternative construction of asymmetric hat basis functions rather than the equispaced ones proposed in (3.2). The main benefits of using the new basis are twofold. First, since the piecewise linearity of the approximation is preserved, the property of satisfying the inequality constraints everywhere in the space holds for the new representation. Second, as the design of the knots is not restricted to be equispaced, one may prefer adding knots only in places where a higher quality of representation is required while preserving the tensor structure. For the latter, we introduce a sequential algorithm for the automatic knot insertion using an evolution criterion based on the maximisation of the integrated squared error of the MAP estimate (iMAP-SE criterion). We test the performance of the proposed sequential algorithm on various synthetic examples up to 5D.

5.2 Approximation of Gaussian processes with asymmetric hat basis functions

For ease of reading, we first consider the 1D finite-dimensional representation discussed in Chapter 3. We adapt the model in (3.1) for the case of non-equispaced design of the knots. We then introduce a sequential algorithm for placing knots preferentially in regions requiring a higher quality of representation. However, the extension to higher dimensions (i.e. $d \geq 2$) can be achieved by tensorisation (see Section 5.3 for more details).

5.2.1 1D construction of the asymmetric hat basis functions

Consider the 1D finite-dimensional representation Y_m from (3.1). Unlike Section 3.2, we consider a (sorted) non-equispaced set of knots $0 = t_1 < \cdots < t_m = 1$ and inter-spaces $\Delta_{m,j} = t_{j+1} - t_j$ for $j = 1, \cdots, m-1$. Then, for a given $x \in [t_{j-1},\ t_{j+1}]$ for $j = 2, \cdots, m-1$, the asymmetric hat basis function $\phi_j(x)$ can be written as

$$\phi_j(x) := \begin{cases} \frac{x - t_{j-1}}{t_j - t_{j-1}} & \text{if } t_{j-1} \leq x < t_j, \\ \frac{t_{j+1} - x}{t_{j+1} - t_j} & \text{if } t_j \leq x \leq t_{j+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

Figure 5.1 illustrates the effect of the asymmetric hat basis functions for the example in Figure 3.1. In contrast to Figure 3.1, here one can observe that the new asymmetric basis gives a non-equispaced design of the knots leading to a more flexible approximation Y_m. One must also note that, since the piecewise linearity holds using (5.1), the properties of the finite-dimensional approximation from Chapter 3 are preserved (e.g. ensuring the inequality constraints everywhere).

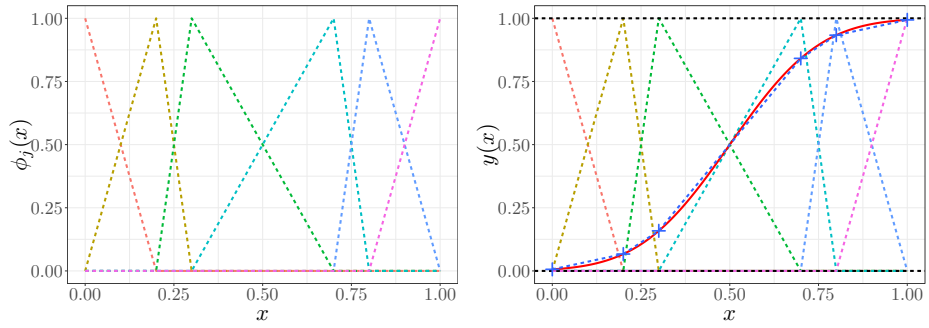


Figure 5.1: Illustration of the finite-dimensional approximation in Figure 3.1 using the asymmetric hat basis functions from (5.1). (Left) Asymmetric hat functions ϕ_j for $j = 1, \cdots, 6$. (Right) Approximation of the function $y(x)$. Red solid and blue dashed lines are the function y, and its finite approximation with six knots given by blue crosses, respectively. Horizontal black dashed lines denote the bounds.

5.2.2 Sequential algorithm for the knot insertion via integrated MAP squared error criterion

Note that it is worth preferentially placing knots only in regions requiring high quality of resolution since the complexity of the approximation in (3.1) increases as m gets larger. This problem is strongly related to free-knot spline approximations where only the most "influential" knots are used (Creutzig et al., 2007; De Boor, 2001, 2002; Goldman, 2003; Hu, 1993; Jupp, 1978; Kobbelt, 2002; Slassi, 2014). Those sets of knots are chosen given diverse criteria. Many of those approaches consider a fixed value of m, and according to an optimal decision rule, sequential algorithms are applied aiming at removing the "less important" knots. In practice, decision rules are based on the minimisation of the linearised bending energy (De Boor, 2001):

$$E(f) := \int_a^b |f(x) - f_m(x)|^2 dx, \tag{5.2}$$

where $x \in [a, b]$, and f_m is the spline approximation of an arbitrary continuous function f on $\mathbb{R}$ at a given set of knots $t_1, \cdots, t_m$. By minimising (5.2), we aim at removing knots such that f_m is able to accurately approximate f. Note that it is assumed in (5.2) that $f(x)$ can be evaluated for any $x \in [a, b]$. However, evaluations of f are not always feasible or involve expensive procedures. When the approximation f_m becomes costly-to-evaluate for large values of m, one would prefer adding "influential" knots rather than removing the less important ones from predefined sets. We refer to, e.g., (Creutzig et al., 2007; De Boor, 2002; Goldman, 2003; Jupp, 1978; Kobbelt, 2002) for a further discussion on knot insertion techniques in spline interpolation, their numerical and asymptotic convergences. In our case, since the proposed finite-dimensional GP Y_m in (3.1) becomes costly-to-evaluate for large values of m, we focus on a free-knot approach based on knot insertions. One must also note that our target function f is assumed to be unknown but evaluations $f(x_1), \cdots, f(x_n)$ are available.

We now make use of the property in Theorem 3.1. As the MAP estimate Y_m^{MAP} of the truncated process in (3.6) converges to the spline interpolation, one may think on adding knots regarding this convergence. Note also that the posterior mean could also be considered but it results in costly (or if not intractable) procedures since the approximation of the entire posterior distribution is required.

Consider the notation in Chapters 3 and 4. Let $Y_{\mathcal{S}_i}$ be the piecewise linear interpolation of the GP Y at a set of knots, $\mathcal{S}_i = \{(t_1, \cdots, t_{m_i}) \text{ s.t. } 0 = t_1 < \cdots < t_{m_i} = 1\}$:

$$Y_{\mathcal{S}_i}(x) = \sum_{j=1}^{m_i} \xi_j \phi_{i,j}(x), \tag{5.3}$$

where $\boldsymbol{\xi}_i = [\xi_1, \cdots, \xi_{m_i}]^\top$ is a centred Gaussian vector with covariance matrix $\boldsymbol{\Gamma}_i = (k(t_j, t_{j'}))_{1 \leq j,j' \leq m_i}$, and $\phi_{i,1} \cdots, \phi_{i,m_i}$ are asymmetric hat basis functions given by (5.1). Consider adding a new knot t_* to the representation in (5.3), such that $t_\nu < t_* < t_{\nu+1}$ for $\nu = 1, \cdots, m_i - 1$. This leads to a finer approximation $Y_{\mathcal{S}_{i+1}}$ given by

$$Y_{\mathcal{S}_{i+1}}(x) = \sum_{j=1}^{m_i} \xi_j \phi_{i+1,j}(x) + \xi_* \phi_{i+1,\nu_*}(x), \tag{5.4}$$

with $m_{i+1} = m_i + 1$, $\xi_* := Y(t_*)$, $\mathcal{S}_{i+1}$ the new set of knots after adding t_*, ν_* the index in $\mathcal{S}_{i+1}$ corresponding to t_*, and $\phi_{i+1,\cdot}$ the new set of basis functions after adding t_*.

In order to properly place the knot t_*, we consider the integrated MAP squared error (iMAP-SE) between the approximations $Y_{\mathcal{S}_i}$ and $Y_{\mathcal{S}_{i+1}}$:

$$\text{iMAP-SE}(t_*) = \int_0^1 [Y_{\mathcal{S}_{i+1}}^{\text{MAP}}(x) - Y_{\mathcal{S}_i}^{\text{MAP}}(x)]^2 dx. \tag{5.5}$$

Then, t_* can be chosen regarding the maximisation of the iMAP-SE criterion, i.e.

$$t_*^{\text{opt}} \in \arg\max_{t_*} \ \text{iMAP-SE}(t_*). \tag{5.6}$$

Note that, in contrast to the optimisation problem in (5.2), by maximising (5.5) we aim at adding a knot t_* such that the finer MAP estimate $Y_{\mathcal{S}_{i+1}}^{\text{MAP}}$ differs as much as possible from $Y_{\mathcal{S}_i}^{\text{MAP}}$. This means that we are looking for convergence of Y_m^{MAP} after having no variations when adding new knots. Then, one can stop the knot insertion algorithm until convergence of the MAP estimate or if the iMAP-SE criterion is smaller than a fixed tolerance $\epsilon \in \mathbb{R}^+$.

We now focus on the computation of (5.5). Let $\boldsymbol{\mu}_i^{\text{M}}$ be the posterior mode given by minimising the quadratic problem,

$$\boldsymbol{\mu}_i^M = \min\{\boldsymbol{\xi}_i^\top \boldsymbol{\Gamma}_i^{-1} \boldsymbol{\xi}_i, \ \boldsymbol{\Phi}_i \boldsymbol{\xi}_i = \boldsymbol{y}, \boldsymbol{\xi}_i \in \mathcal{C}_i\}, \quad \text{for} \quad i \in \mathbb{N},$$

with $\boldsymbol{\Phi}_i$ the $n \times m_i$ matrix defined by $(\boldsymbol{\Phi}_i)_{l,j} = \phi_{i,j}(x_l)$, and $\mathcal{C}_i$ a convex set of $\mathbb{R}^{m_i}$ corresponding to the inequality constraints. Then,

$$Y_{\mathcal{S}_i}^{\text{MAP}}(x) = \sum_{j=1}^{m_i} \mu_{i,j}^M \phi_{i,j}(x), \tag{5.7}$$

with $\mu^M_{i,j}$ the j-th component of $\boldsymbol{\mu}^M_i$. Using (5.4) and (5.7), we have

$$Y^{\text{MAP}}_{\mathcal{S}_{i+1}}(x) - Y^{\text{MAP}}_{\mathcal{S}_i}(x) = \sum_{j=1}^{m_i} \mu^M_{i+1,j}\phi_{i+1,j}(x) + \mu^M_{i+1,\nu_*}\phi_{i+1,\nu_*}(x) - \sum_{j=1}^{m_i} \mu^M_{i,j}\phi_{i,j}(x), \tag{5.8}$$

Let the vectors of coefficients $\boldsymbol{\beta}_i = [\mu^M_{i,1},\ \cdots,\ \mu^M_{i,\nu},\ \mu^M_{i,\nu+1},\ \cdots,\ \mu^M_{i,m_i}]^\top \in \mathbb{R}^{m_i}$ and $\boldsymbol{\beta}_{i+1} = [\mu^M_{i+1,1},\ \cdots,\ \mu^M_{i+1,\nu},\ \mu^M_{i+1,\nu_*},\ \mu^M_{i+1,\nu+1},\ \cdots,\ \mu^M_{i+1,m_i}]^\top \in \mathbb{R}^{m_{i+1}}$. Then, (5.8) can be matricially written as

$$Y^{\text{MAP}}_{\mathcal{S}_{i+1}}(x) - Y^{\text{MAP}}_{\mathcal{S}_i}(x) = \boldsymbol{\beta}^\top_{i+1}\boldsymbol{\phi}_{i+1}(x) - \boldsymbol{\beta}^\top_i\boldsymbol{\phi}_i(x), \tag{5.9}$$

where $\boldsymbol{\phi}_{i+1}(x) = [\phi_{i+1,1}(x),\ \cdots,\ \phi_{i+1,\nu}(x),\ \phi_{i+1,\nu_*}(x),\ \phi_{i+1,\nu+1}(x),\ \cdots,\ \phi_{i+1,m_i}(x)]^\top$ and $\boldsymbol{\phi}_i(x) = [\phi_{i,1}(x),\ \cdots,\ \phi_{i,\nu}(x),\ \phi_{i,\nu+1}(x),\ \cdots,\ \phi_{i,m_i}(x)]^\top$. One can note that many of the hat basis functions $\phi_{i,\cdot}$ are exactly equal to the new basis $\phi_{i+1,\cdot}$, except for those related to t_ν and $t_{\nu+1}$. Furthermore, $\boldsymbol{\phi}_{i+1}$ correspond to a finer space of affine functions than $\boldsymbol{\phi}_i$, since there is one additional knot (t_*). Hence, it is possible to write the MAP estimate $Y^{\text{MAP}}_{\mathcal{S}_i}(x) = \boldsymbol{\beta}^\top_i\boldsymbol{\phi}_i(x)$ in terms of the new basis $\boldsymbol{\phi}_{i+1}$. More precisely, we have:

$$\boldsymbol{\beta}^\top_i\boldsymbol{\phi}_i(x) = \boldsymbol{\beta}'^\top_i\boldsymbol{\phi}_{i+1}(x), \tag{5.10}$$

with

$$\boldsymbol{\beta}'_i = [\mu^M_{i,1},\ \cdots,\ \mu^M_{i,\nu},\ \mu^M_{i,\nu}\phi_{i,\nu}(t_*) + \mu^M_{i,\nu+1}\phi_{i,\nu+1}(t_*),\ \mu^M_{i,\nu+1},\ \cdots,\ \mu^M_{i,m_i}]^\top. \tag{5.11}$$

We refer to the proofs in subsection 5.5.1 for further details. Notice that the vector $\boldsymbol{\beta}'_i$ is similar to $\boldsymbol{\beta}_i$ but with the additional term $\mu^M_{i,\nu}\phi_{i,\nu}(t_*) + \mu^M_{i,\nu+1}\phi_{i,\nu+1}(t_*)$ corresponding to value of $Y^{\text{MAP}}_{\mathcal{S}_i}$ at t_*.

Thus, (5.9) can be rewritten as

$$Y^{\text{MAP}}_{\mathcal{S}_{i+1}}(x) - Y^{\text{MAP}}_{\mathcal{S}_i}(x) = [\boldsymbol{\beta}_{i+1} - \boldsymbol{\beta}'_i]^\top\boldsymbol{\phi}_{i+1}(x), \tag{5.12}$$

and the iMAP-SE criterion can be efficiently computed as detailed in the following proposition.

Proposition 5.1 *The* iMAP-SE *criterion in* (5.5) *is given by*

$$\text{iMAP-SE}(t_*) = \boldsymbol{\zeta}^\top_{i+1}\boldsymbol{\Psi}_{i+1}\boldsymbol{\zeta}_{i+1},$$

where $\boldsymbol{\zeta}_{i+1} = \boldsymbol{\beta}_{i+1} - \boldsymbol{\beta}'_i$ *with* $\boldsymbol{\beta}'_i$ *is given by* (5.11)*, and* $\boldsymbol{\Psi}_{i+1} = \int_0^1 \boldsymbol{\phi}_{i+1}(x)\boldsymbol{\phi}^\top_{i+1}(x)dx$ *is the symmetric 3-banded matrix given by*

$$(\boldsymbol{\psi}_{i+1})_{j,j'} = \begin{cases} \frac{t_{j+1}-t_j}{3} & \text{if } j = j' = 1, \\ \frac{t_j-t_{j-1}}{3} & \text{if } j = j' = m_i, \\ \frac{t_{j+1}-t_{j-1}}{3} & \text{if } j = j' = 2, \cdots, m_i - 1, \\ \frac{t_{j+1}-t_j}{6} & \text{if } j' = j+1,\ j = 1, \cdots, m_i - 1, \\ 0 & \text{if } |j - j'| > 1. \end{cases}$$

Algorithm 2 1D sequential algorithm for the automatic knot insertion based on the iMAP-SE evolution criterion.

1: **Input:** Initial set of knots $m_0 \in \mathbb{N}$ and tolerance parameter $\epsilon \in \mathbb{R}^+$.
2: Define $Y_{\mathcal{S}_0} = \sum_{j=1}^{m_0} \xi_j \phi_{0,j}(x)$, with asymmetric hat basis functions $\phi_{0,\cdot}$ from (5.1).
3: Compute $Y_{\mathcal{S}_0}^{\text{MAP}}$ subject to both interpolation and inequality constraints as in (5.7).
4: **for** $i = 1, 2, \cdots$ **do**
5: Add a new knot t_* such that the iMAP-SE criterion in (5.5) is maximised, i.e.

$$t_*^{\text{opt}} \in \arg\max_{t_*} \int_0^1 [Y_{\mathcal{S}_{i+1}}^{\text{MAP}}(x) - Y_{\mathcal{S}_i}^{\text{MAP}}(x)]^2 dx,$$

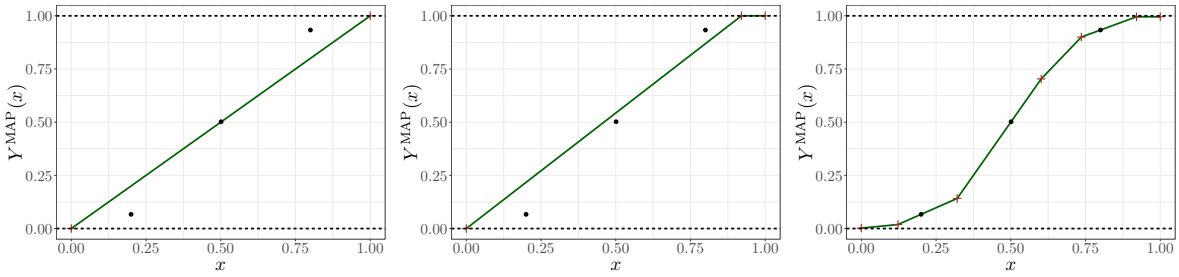
with $Y_{\mathcal{S}_{i+1}}$ as in (5.4).
6: **if** iMAP-SE$(t_*^{\text{opt}}) < \epsilon$ **then** stop.

Without considering the complexity required for the computation of the vectors $\boldsymbol{\beta}_i$ and $\boldsymbol{\beta}_{i+1}$ (that are obtained by solving strictly convex quadratic programs), then computing the iMAP-SE criterion in (5.5) has a complexity of $\mathcal{O}(m_{i+1})$. Finally, a sequential routine for the automatic knot insertion can be performed (see Algorithm 2). Note that, as stopping rule, we check the iMAP-SE criterion at t_*^{opt} until having a value smaller than a given tolerance parameter $\epsilon \in \mathbb{R}^+$.

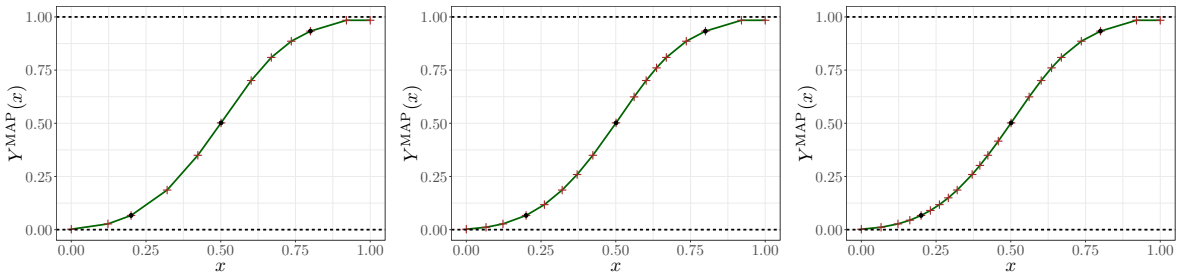
5.2.3 Numerical illustration

We illustrate the performance of Algorithm 2 using the 1D toy example in Figure 3.2(d). We initialise the sequential algorithm with knots only at the boundaries, i.e. at $t_1 = 0$ and $t_2 = 1$. This leads to a model with linear trajectories in $[0, 1]$. One must note that this initial representation is not of interest for practical implementations but is helpful for illustrations. As in Figure 3.2(d), we use a SE kernel with fixed parameters $(\sigma^2 = 1,\ \ell = 0.2)$, and consider an additive Gaussian white noise with variance $\tau^2 = 10^{-5}$. Here, we fix a tolerance parameter $\epsilon = 10^{-3}$

Figures 5.2 and 5.3 show the evolution of the MAP estimate and the conditional sample-path under both boundedness and monotonicity constraints. Note that the knots are placed in regions where the target function is more likely to exhibit high variations. In particular, the algorithm focused on the insertion of knots around $x = 0.25$ due to the influence of the lower bound $l = 0$. After the first 10 iterations, one can observe that the algorithm starts to exhibit convergence of the MAP estimate Y_m^{MAP}, with smoother predictions after the 15th iteration. Compared to Figure 3.2(d), one can note that a similar profile is obtained in Figure 5.3 using only $m = 20$ rather than using the 100 (equispaced) knots suggested in Figure 3.2(d). We used the Q^2 and coverage accuracy (CA) criteria to assess the quality of predictions over the 50 new values. We refer to (3.17) for the definition of the Q^2 criterion. Denoting by n_t the number of test points, and $\widehat{z}_1, \cdots, \widehat{z}_{n_t}$ the sets of predicted observations, then the CA assesses the quality of predictive variances $\widehat{\sigma}_i^2$ for $i = 1, \cdots, n_t$. Here, we use one standard deviation intervals $(\widehat{z}_i \pm \widehat{\sigma}_i)$



(a) $i = 0$ (b) $i = 1, \text{iMAP-SE} = 2.1$ (c) $i = 5, \text{iMAP-SE} = 0.6$



(d) $i = 10, \text{iMAP-SE} = 9.8 \times 10^{-2}$ (e) $i = 15, \text{iMAP-SE} = 7.6 \times 10^{-3}$ (f) $i = 20, \text{iMAP-SE} = 4.7 \times 10^{-5}$

Figure 5.2: MAP evolution of the 1D toy example from subsection 5.2.3 after i iterations of the sequential algorithm. Each panel shows: training points and knots (black dots and red crosses), the MAP estimate (green solid line), and the resulting iMAP-SE values (subcaption). Horizontal black dashed lines denote the bounds.

which should provide a pointwise coverage of the test data around 68%. Departure from $\text{CA}_{\pm\sigma} = 0.68$ may indicate that the confidence intervals are too large (respectively small) for coverage values of $\text{CA}_{\pm\sigma} > 0.68$ (resp. $\text{CA}_{\pm\sigma} < 0.68$) of the predictive variances. We obtained $Q^2 = 0.998$ and $\text{CA}_{\pm\sigma} = 0.661$ values which are competitive with the ones obtained in Figure 3.2(d) (around $Q^2 = 0.999$ and $\text{CA}_{\pm\sigma} = 0.670$).

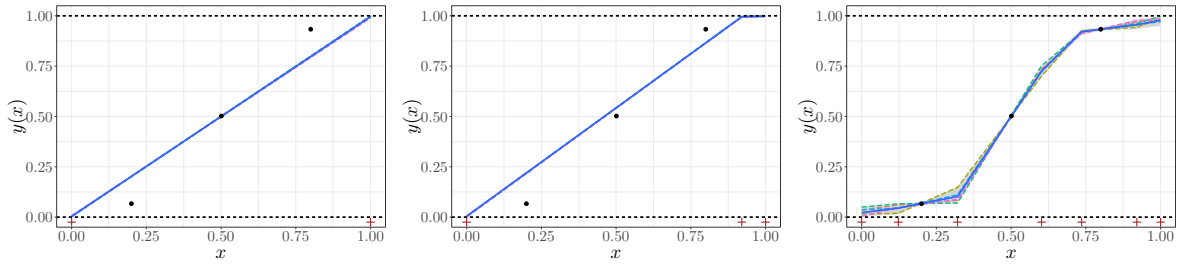
5.3 Extension to high dimensions

The construction of the asymmetric hat basis functions in (5.1) and the iMAP-SE criterion in (5.5) can be extended to higher dimensions (i.e. $d \geq 2$) by tensorisation. Next, we focus on the computation of the iMAP-SE in dimension d.

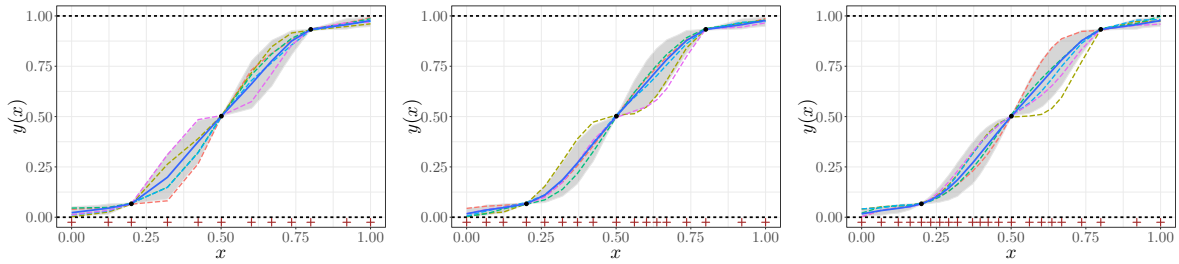
5.3.1 Integrated MAP squared error in dimension d

Let the set of (ordered) knots per dimension $(t_1^1, \cdots, t_{m_1}^1), \cdots, (t_1^d, \cdots, t_{m_d}^d)$. Consider the initial finite representation $Y_{\mathcal{S}_i}$ given by

$$Y_{\mathcal{S}_i}(\boldsymbol{x}) = \sum_{j_1=1}^{m_{i,1}} \cdots \sum_{j_d=1}^{m_{i,d}} \xi_{j_1,\cdots,j_d} \prod_{\kappa=1}^{d} \phi_{i,j_\kappa}^{\kappa}(x_\kappa), \tag{5.13}$$



(a) $i = 0$ (b) $i = 1$, iMAP-SE $= 2.1$ (c) $i = 5$, iMAP-SE $= 0.6$



(d) $i = 10$, iMAP-SE $= 9.8 \times 10^{-2}$ (e) $i = 15$, iMAP-SE $= 7.6 \times 10^{-3}$ (f) $i = 20$, iMAP-SE $= 4.7 \times 10^{-5}$

Figure 5.3: Conditional sample-path evolution of the example in Figure 5.2. Each panel shows: training points and knot locations (black dots and red crosses), the conditional mean function (blue solid line), the 90% confidence intervals (grey region), and conditional realisations (dashed lines). Horizontal black dashed lines denote the bounds.

with $\boldsymbol{x} = (x_1, \cdots, x_d) \in [0,1]^d$. Observe that the vector $\boldsymbol{\xi}_i = [\xi_{1,\cdots,1}, \ \cdots, \ \xi_{m_{i,1},\cdots,m_{i,d}}]^\top \in \mathbb{R}^{m_{i,1}\times\cdots\times m_{i,d}}$ of the values at the knots is a centred Gaussian vector with covariance matrix $\boldsymbol{\Gamma}_i$. Without loss of generality, we consider adding a new row of knots at $t^1_\nu < t^1_* < t^1_{\nu+1}$ to the initial approximation in (5.13) along the first input dimension x_1. This leads to a finer approximation $Y_{\mathcal{S}_{i+1}}$ given by

$$Y_{\mathcal{S}_{i+1}}(\boldsymbol{x}) = \sum_{j_1=1}^{m_{i,1}} \cdots \sum_{j_d=1}^{m_{i,d}} \xi_{j_1,\cdots,j_d} \prod_{\kappa=1}^{d} \phi^\kappa_{i+1,j_\kappa}(x_\kappa) + \sum_{j_2=1}^{m_{i,2}} \cdots \sum_{j_d=1}^{m_{i,d}} \xi_{\nu^*_1,\cdots,j_d} \phi^1_{i+1,\nu^*_1}(x_1) \prod_{\kappa=2}^{d} \phi^\kappa_{i+1,j_\kappa}(x_\kappa), \tag{5.14}$$

where $m_{i+1} = m_i + 1$, $\xi_{\nu^*_1,\cdots,j_d} := Y(t^1_*, \cdots, t^d_{j_d})$, ν^*_1 the index in $\mathcal{S}_{i+1}$ corresponding to t^1_*, and the $\phi^1_{i+1,\cdot}$ correspond to the new set of asymmetric hat basis functions after adding the row of knots at t^1_*.

Then, the iMAP-SE criterion can be written as

$$\text{iMAP-SE}(t^1_*) = \int_0^1 \cdots \int_0^1 [Y^{\text{MAP}}_{\mathcal{S}_{i+1}}(\boldsymbol{x}) - Y^{\text{MAP}}_{\mathcal{S}_i}(\boldsymbol{x})]^2 dx_1 \cdots dx_d, \tag{5.15}$$

and

$$t^{1,\text{opt}}_* \in \arg\max_{t^1_*} \ \text{iMAP-SE}(t^1_*). \tag{5.16}$$

As in subsection 5.2.2, that criterion can be computed efficiently. Let $\boldsymbol{\mu}_i^{\mathrm{M}} \in \mathbb{R}^{m_{i,1}\times\cdots\times m_{i,d}}$ and $\boldsymbol{\mu}_{i+1}^{\mathrm{M}} \in \mathbb{R}^{m_{i+1,1}\times\cdots\times m_{i,d}}$ be the posterior modes of (5.13) and (5.14), respectively. Then, we have

$$Y_{\mathcal{S}_i}^{\mathrm{MAP}}(\boldsymbol{x}) = \sum_{j_1=1}^{m_{i,1}} \cdots \sum_{j_d=1}^{m_{i,d}} \mu_{i,j_1,\cdots,j_d}^{M} \prod_{\kappa=1}^{d} \phi_{i,j_\kappa}^{\kappa}(x_\kappa) = \boldsymbol{\beta}_i^{\top} \boldsymbol{\Phi}_i(\boldsymbol{x}), \tag{5.17}$$

with $\boldsymbol{\beta}_i = [\mu_{i,1,\cdots,1}^{M},\ \cdots, \mu_{i,m_{i,1},\cdots m_{i,d}}^{M}] \in \mathbb{R}^{m_{i,1}\times\cdots\times m_{i,d}}$, $\boldsymbol{\Phi}_i(\boldsymbol{x}) = \prod_{\kappa=1}^{d} \boldsymbol{\phi}_{i,\kappa}(x_\kappa)$ and

$$\boldsymbol{\phi}_{i,\kappa}(x_\kappa) = [\phi_{i,1}^{\kappa}(x_\kappa),\ \cdots,\ \phi_{i,m_{i,\kappa}}^{\kappa}(x_\kappa)]^{\top}.$$

Similarly for $Y_{\mathcal{S}_{i+1}}^{\mathrm{MAP}}$, we have

$$\begin{aligned} Y_{\mathcal{S}_{i+1}}^{\mathrm{MAP}}(\boldsymbol{x}) &= \sum_{j_1=1}^{m_{i,1}} \cdots \sum_{j_d=1}^{m_{i,d}} \mu_{i+1,j_1,\cdots,j_d}^{M} \prod_{\kappa=1}^{d} \phi_{i+1,j_\kappa}^{\kappa}(x_\kappa) \\ &\quad + \sum_{j_2=1}^{m_{i,2}} \cdots \sum_{j_d=1}^{m_{i,d}} \mu_{i+1,j_1=\nu_1^*,\cdots,j_d}^{M} \phi_{i+1,j_1=\nu_1^*}^{1}(x_1) \prod_{\kappa=2}^{d} \phi_{i+1,j_\kappa}^{\kappa}(x_\kappa) \\ &= \boldsymbol{\beta}_{i+1}^{\top} \boldsymbol{\Phi}_{i+1}(\boldsymbol{x}), \end{aligned} \tag{5.18}$$

where $\boldsymbol{\Phi}_{i+1}(\boldsymbol{x}) = \prod_{\kappa=1}^{d} \boldsymbol{\phi}_{i+1,\kappa}(x_\kappa)$, and the vector $\boldsymbol{\beta}_{i+1} \in \mathbb{R}^{(m_{i,1}+1)\times\cdots\times m_{i,d}}$ is given by $\boldsymbol{\beta}_{i+1} = [\mu_{i+1,1,\cdots,1}^{M},\ \cdots,\ \mu_{i+1,\nu_1^*,\cdots,1}^{M},\ \cdots,\ \mu_{i+1,\nu_1^*,\cdots,m_{i,d}}^{M},\ \cdots,\ \mu_{i+1,m_{i,1},\cdots,m_{i,d}}^{M}]$. As discussed in subsection 5.2.2 for 1D implementations, it is possible to write the hat basis functions $\phi_{i,\cdot}$ in terms of the new basis $\phi_{i+1,\cdot}$:

$$\boldsymbol{\beta}_i^{\top} \boldsymbol{\Phi}_i(\boldsymbol{x}) = \boldsymbol{\beta}_i'^{\top} \boldsymbol{\Phi}_{i+1}(\boldsymbol{x}), \tag{5.19}$$

with the vector $\boldsymbol{\beta}_i' \in \mathbb{R}^{(m_{i,1}+1)\times\cdots\times m_{i,d}}$ defined as in (5.31) in the proofs from subsection 5.5.2. Thus, the difference of the MAP estimates is given by

$$Y_{\mathcal{S}_{i+1}}^{\mathrm{MAP}}(\boldsymbol{x}) - Y_{\mathcal{S}_i}^{\mathrm{MAP}}(\boldsymbol{x}) = [\boldsymbol{\beta}_{i+1} - \boldsymbol{\beta}_i']^{\top} \boldsymbol{\Phi}_{i+1}(\boldsymbol{x}),$$

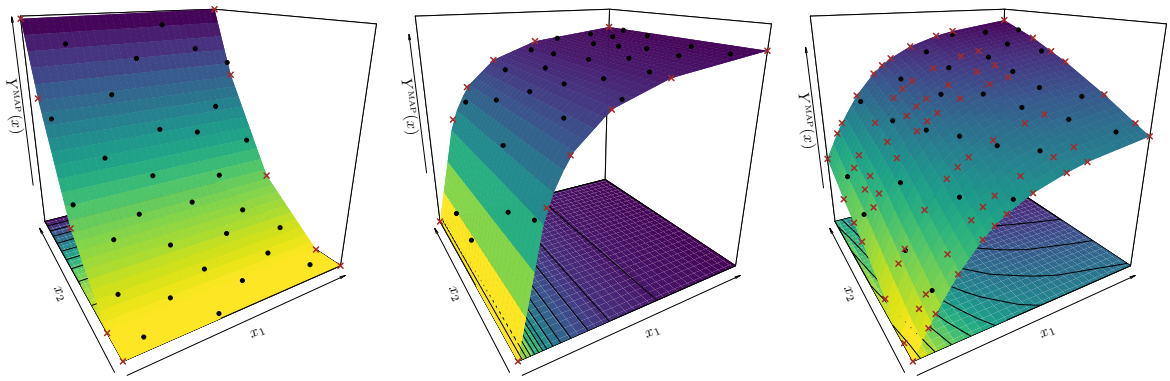
with $\boldsymbol{\Phi}_{i+1}(\boldsymbol{x}) = \prod_{\kappa=1}^{d} \boldsymbol{\phi}_{i+1,\kappa}(x_\kappa)$.

Proposition 5.2 *The* iMAP-SE *criterion of* (5.15) *is given by*

$$\text{iMAP-SE}(t_*^1) = \boldsymbol{\zeta}_{i+1}^{\top} \boldsymbol{\Psi}_{i+1} \boldsymbol{\zeta}_{i+1},$$

with $\boldsymbol{\zeta}_{i+1} = \boldsymbol{\beta}_{i+1} - \boldsymbol{\beta}_i'$ *where* $\boldsymbol{\beta}_i'$ *is given by* (5.31), $\boldsymbol{\Psi}_{i+1} = \bigotimes_{\kappa=1}^{d} \widetilde{\boldsymbol{\Psi}}_{i+1,\kappa}$, *and* $\widetilde{\boldsymbol{\Psi}}_{i+1,\cdot}$ *symmetric 3-banded matrices per dimension as in Proposition 5.1.*

Finally, the sequential scheme in Algorithm 2 can be used by checking (5.16) for each dimension and choosing the one leading the highest iMAP-SE value.



(a) Toy example $y_1(x_1, x_2)$ (b) Toy example $y_2(x_1, x_2)$ (c) Toy example $y_3(x_1, x_2)$

Figure 5.4: MAP estimate of the 2D examples in (5.20) after convergence of the sequential algorithm. Each panel shows: training points and knots (black dots and red crosses), and the MAP estimate (solid surface).

Table 5.1: iMAP-SE results for the examples in Figure 5.4. The number of iterations N_{iter} and number of knots (m_1, m_2) are also shown.

Example	N_{iter}	iMAP-SE	m_1	m_2
Figure 5.4(a)	3	3.06×10^{-2}	2	5
Figure 5.4(b)	4	8.24×10^{-3}	6	2
Figure 5.4(c)	14	5.24×10^{-3}	11	7

5.3.2 Numerical illustration in 2D

Here, we test the performance of the sequential algorithm on various 2D toy examples. We analyse two cases: target functions exhibiting inactive dimensions and full variable functions. As target functions, we consider:

$$\begin{aligned} y_1(x_1, x_2) &= x_2^2, \\ y_2(x_1, x_2) &= \arctan(10x_1), \\ y_3(x_1, x_2) &= \arctan(5x_1) + \arctan(x_2), \end{aligned} \tag{5.20}$$

with $(x_1, x_2) \in [0, 1]^2$. We train GP models with monotonicity constraints using 2D SE kernels with parameters $\boldsymbol{\theta} = (\sigma^2, \ell_1, \ell_2)$ estimated via ML. We initialise the sequential algorithm using knots at the corners of the unit square. We train the GP models using fixed maximin Latin hypercube DoEs at 30 locations.

Figure 5.4 shows the MAP estimates for target functions in (5.20) under monotonicity constraints. Observe that the algorithm can properly identify which input dimensions are more active, and it adds knots over those ones requiring higher resolution of representation. In Table 5.1, we show the total number of iterations required for having convergence

Table 5.2: iMAP-SE evolution results for the monotonic example in Figure 4.5.

Iteration	Knots' evolution					iMAP-SE	t_*^{opt}	CPU Time [min]
0	2	2	2	2	2	—	—	—
1	**1**	0	0	0	0	4.28×10^{-2}	$t_*^{1,\text{opt}} = 0.233$	2.67
2	0	0	0	0	**1**	1.19×10^{-2}	$t_*^{5,\text{opt}} = 0.229$	4.51
3	0	0	0	0	**1**	1.66×10^{-2}	$t_*^{5,\text{opt}} = 0.741$	4.23
4	0	0	0	**1**	0	8.96×10^{-3}	$t_*^{4,\text{opt}} = 0.442$	4.20
5	**1**	0	0	0	0	9.18×10^{-3}	$t_*^{1,\text{opt}} = 0.550$	4.21
6	0	**1**	0	0	0	1.16×10^{-3}	$t_*^{2,\text{opt}} = 0.400$	4.03
7	0	0	0	**1**	0	5.60×10^{-4}	$t_*^{4,\text{opt}} = 0.661$	3.82
8	0	0	0	**1**	0	5.12×10^{-4}	$t_*^{4,\text{opt}} = 0.250$	3.84
9	0	0	0	0	**1**	3.60×10^{-4}	$t_*^{5,\text{opt}} = 0.345$	4.34
10	0	0	0	0	**1**	2.08×10^{-4}	$t_*^{5,\text{opt}} = 0.615$	4.73
11	**1**	0	0	0	0	1.56×10^{-4}	$t_*^{1,\text{opt}} = 0.382$	5.06
12	**1**	0	0	0	0	4.84×10^{-4}	$t_*^{1,\text{opt}} = 0.870$	6.02
13	0	**1**	0	0	0	1.73×10^{-4}	$t_*^{2,\text{opt}} = 0.686$	7.81
14	0	0	**1**	0	0	1.16×10^{-4}	$t_*^{3,\text{opt}} = 0.481$	11.34
15	**1**	0	0	0	0	4.22×10^{-5}	$t_*^{1,\text{opt}} = 0.691$	21.40
Number of knots	**7**	**4**	**3**	**5**	**6**			92.21

of the sequential algorithm, and we also display the resulting iMAP-SE values and knot insertion per dimension.

5.3.3 5D toy example under monotonicity constraints

We consider the 5D toy example in subsection 4.2.3 under monotonicity constraints. There, noisy evaluations from (4.6) have been taken on a maximin Latin-hypercube DoE over $[0,1]^5$ at 10^2 locations. We here assume that there is no prior information above the behaviour of the target function in (4.6). We initialise the sequential algorithm using knots at the corners of the unit hyper-square $[0,1]^5$, leading to an initial number of 10 knots. Aiming at testing only the computational cost of the algorithm, we assume that the covariance parameters were known and equal to the ones obtained in subsection 4.2.3. The tolerance parameter is fixed to $\epsilon = 10^{-4}$.

The evolution of the sequential construction is displayed in Table 5.2. One can observe that, as suggested in subsection 4.2.3, less knots were required along the third dimension due to the linearity of the target function y w.r.t. x_3. Furthermore, the algorithm focused on the first and fifth dimensions since y varies more across this dimension compared to the other ones. After convergence of the algorithm, we obtained the set of knots per dimension: $(0, 0.23, 0.38, 0.55, 0.69, 0.87, 1)$, $(0, 0.4, 0.69, 1)$, $(0, 0.48, 1)$, $(0, 0.25, 0.44, 0.66, 1)$ and $(0, 0.23, 0.35, 0.62, 0.74, 1)$.

Results in Table 5.2 were obtained after a lapse of 1.5 hours on five cores of an Intel® Core™ i7-6700HQ CPU, i.e. a single core for solving (5.16) for each dimension. This execution time can be reduced by adding more knots per iteration rather than adding a single one. Notice also that, for illustrative purposes, the sequential algorithm has been initialised with knots at the corners of the compact space $[0,1]^5$. In practice, it is worth to properly initialise the sets of knots by taking into account the dynamics of functions.

5.4 Conclusions

We explored an alternative construction of asymmetric hat basis functions for the finite-dimensional approximation in Chapters 3 and 4. The main benefit of using such asymmetric construction is that they allow us to preferentially insert knots only in regions requiring high quality of representation (typically in highly variable regions), while preserving the properties of the approximation in Chapter 3 (e.g. satisfying the inequality constraints everywhere).

We also introduced a dedicated sequential algorithm for the automatic knot insertion regarding convergence of the MAP estimate of the constrained process. We proposed an evolution criterion based on the integrated MAP squared error (iMAP-SE) in order to preferentially adding knots in regions yielding the maximum variation between consecutive MAP estimates. We derived closed-form formulas for the iMAP-SE, corresponding to a small computational cost. As shown on various synthetic examples up to 5D, the proposed sequential algorithm allowed to refine the grid of knots only across those dimensions demanding a better quality of resolution.

The efficiency of the sequential algorithm based on the iMAP-SE criterion has been tested on various numerical illustrations. We believe that, as the number of knots goes to infinity, the rectangular design of knots provided by the algorithm will be dense on the input domain. From the fixed domain asymptotics' point of view, this assumption implies that the asymptotic properties of the finite-dimensional approximation obtained by the symmetric hat basis functions hold for the asymmetric ones (e.g. convergence of the MAP estimate to the spline interpolation as shown by Bay et al., 2016). We could not provide this proof at the moment of writing this manuscript, but it is of our interest as a future work. Furthermore, alternative MAP-based criteria (e.g. integrated MAP absolute error) can be further investigated in future contributions.

5.5 Proof – Integrated MAP squared error criterion

In this section, we write down all the expression required for the computation of the iMAP-SE criterion in 1D and in high dimensions. For the case when $d \geq 2$, due to the tensor structure of the iMAP-SE, we refer to expressions obtained in 1D.

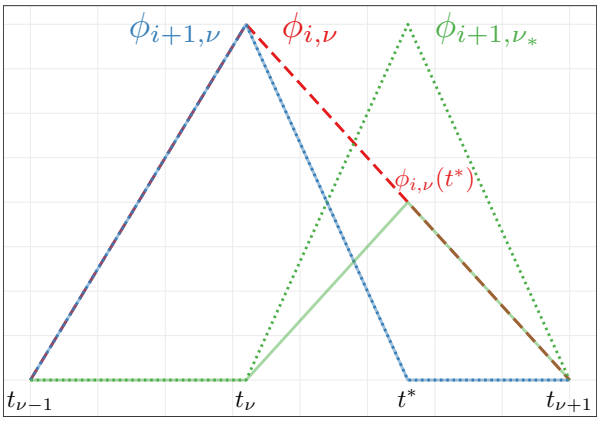


Figure 5.5: Representation of the hat basis function $\phi_{i,\nu}$ (red dashed line) in the vector space spanned by $\phi_{i+1,\nu}$ and ϕ_{i+1,ν_*} (blue and green dotted lines, respectively).

5.5.1 Proof in 1D

5.5.1.A. Link between $\phi_{i,\cdot}$ and $\phi_{i+1,\cdot}$

As the discretisation corresponding to the hat basis function $\boldsymbol{\phi}_{i+1}$ is finer than for $\boldsymbol{\phi}_i$, we can express the latter basis as a function of the former ones. Consider the hat basis functions $\boldsymbol{\phi}_i(x) = [\phi_{i,1}(x),\ \cdots,\ \phi_{i,\nu}(x),\ \phi_{i,\nu+1}(x),\ \cdots,\ \phi_{i,m_i}(x)]^\top$ and the vector $\boldsymbol{\beta}_i = [\mu^M_{i,1},\ \cdots,\ \mu^M_{i,\nu},\ \mu^M_{i,\nu+1},\ \cdots,\ \mu^M_{i,m_i}]^\top \in \mathbb{R}^{m_i}$. Consider also the new set of basis $\boldsymbol{\phi}_{i+1}(x) = [\phi_{i+1,1}(x),\ \cdots,\ \phi_{i+1,\nu}(x),\ \phi_{i+1,\nu_*}(x),\ \phi_{i+1,\nu+1}(x),\ \cdots,\ \phi_{i+1,m_i}(x)]^\top$. Here, we aim at expressing the product $\boldsymbol{\beta}_i^\top \boldsymbol{\phi}_i(x)$ in terms of $\boldsymbol{\phi}_{i+1}(x)$.

As discussed in subsection 5.2.2, many of the hat basis functions $\phi_{i,\cdot}$ are equal to the new basis $\phi_{i+1,\cdot}$, except the ones associated to the knots t_ν and $t_{\nu+1}$, i.e.

$$\phi_{i,j} = \phi_{i+1,j} \quad \text{for} \quad j \neq \nu, \nu+1. \tag{5.21}$$

For the cases when $j = \nu$ and $j = \nu+1$, $\phi_{i,j}$ can be expressed as a linear combinations of the new basis functions $\phi_{i+1,j}$ and ϕ_{i+1,ν_*}. Consider first the basis function $\phi_{i,\nu}$. As a piecewise linear function with knots $t_{\nu-1}$, $t_{\nu+1}$, it can be expressed in the vector space spanned by $\phi_{i+1,\nu}$ and ϕ_{i+1,ν_*} (see Figure 5.5):

$$\phi_{i,\nu}(x) = \phi_{i+1,\nu}(x) + \phi_{i,\nu}(t_*)\phi_{i+1,\nu_*}(x), \tag{5.22}$$

which correspond to the piecewise linear function with knots $t_{\nu-1}$, t_*, $t_{\nu+1}$, hence a larger space of functions. Notice that (5.22) can be obtained by evaluating $\phi_{i,\nu}(x)$ at $x = t_\nu$ and $x = t_*$. Since $t_\nu < t_* < t_{\nu+1}$ is fixed, then the constant $\phi_{i,\nu}(t_*)$ is known and is given by $\phi_{i,\nu}(t_*) = \frac{t_{\nu+1}-t_*}{t_{\nu+1}-t_\nu}$ according to the definition in (5.1).

Similarly, for the hat basis function $\phi_{i,\nu+1}$, we obtain

$$\phi_{i,\nu+1}(x) = \phi_{i+1,\nu+1}(x) + \phi_{i,\nu+1}(t_*)\phi_{i+1,\nu_*}(x), \tag{5.23}$$

with $\phi_{i,\nu+1}(t_*) = \frac{t_*-t_\nu}{t_{\nu+1}-t_\nu}$. Finally, replacing (5.22) and (5.23) in (5.7), we have

$$\begin{aligned} Y^{\text{MAP}}_{\mathcal{S}_i}(x) &= \sum_{\substack{j=1 \\ j\neq\nu,\nu+1}}^{m_i} \mu^M_{i,j}\phi_{i+1,j}(x) + \mu^M_{i,\nu}[\phi_{i+1,\nu}(x) + \phi_{i,\nu}(t_*)\phi_{i+1,\nu_*}(x)] \\ &\quad + \mu^M_{i,\nu+1}[\phi_{i+1,\nu+1}(x) + \phi_{i,\nu+1}(t_*)\phi_{i+1,\nu_*}(x)] \\ &= \sum_{\substack{j=1 \\ j\neq\nu,\nu+1}}^{m_i} \mu^M_{i,j}\phi_{i+1,j}(x) + \mu^M_{i,\nu}\phi_{i+1,\nu}(x) \\ &\quad + [\mu^M_{i,\nu}\phi_{i,\nu}(t_*) + \mu^M_{i,\nu+1}\phi_{i,\nu+1}(t_*)]\phi_{i+1,\nu_*}(x) + \mu^M_{i,\nu+1}\phi_{i+1,\nu+1}(x) \\ &= {\boldsymbol{\beta}'_i}^\top \boldsymbol{\phi}_{i+1}(x), \end{aligned}$$

where ${\boldsymbol{\beta}'_i}^\top = [\mu^M_{i,1},\ \cdots,\ \mu^M_{i,\nu},\ \mu^M_{i,\nu}\phi_{i,\nu}(t_*) + \mu^M_{i,\nu+1}\phi_{i,\nu+1}(t_*),\ \mu^M_{i,\nu+1},\ \cdots,\ \mu^M_{i,m_i}]^\top$.

5.5.1.B. Computing the matrix Ψ

Let $\boldsymbol{\Psi}_i$ be the $m_i \times m_i$ symmetric matrix with entries $\boldsymbol{\Psi}_i = (\int_0^1 \phi_{i,j}(x)\phi_{i,j'}(x)dx)_{1\leq j,j'\leq m_i}$. Since only consecutive hat basis functions show overlaps, then, for a given value of $j \in \{1,\cdots,m_i\}$, we have

$$\int_0^1 \phi_{i,j}(x)\phi_{i,j'}(x)dx = \begin{cases} \int_0^1 \phi^2_{i,j}(x)dx & \text{if } j' = j, \\ \int_0^1 \phi_{i,j}(x)\phi_{i,j-1}(x)dx & \text{if } j' = j-1 \ (j \geq 2), \\ \int_0^1 \phi_{i,j}(x)\phi_{i,j+1}(x)dx & \text{if } j' = j+1 \ (j \leq m_i - 1), \\ 0 & \text{otherwise.} \end{cases}$$

Then, due to the symmetry of the integrals, $\boldsymbol{\Psi}_i$ is the symmetric 3-banded matrix:

$$\boldsymbol{\Psi}_i = \begin{bmatrix} \psi^i_{1,1} & \psi^i_{1,2} & 0 & \cdots & 0 & 0 & 0 \\ \psi^i_{1,2} & \psi^i_{2,2} & \psi^i_{2,3} & \cdots & 0 & 0 & 0 \\ 0 & \psi^i_{2,3} & \psi^i_{3,3} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \psi^i_{m_i-2,m_i-2} & \psi^i_{m_i-2,m_i-1} & 0 \\ 0 & 0 & 0 & \cdots & \psi^i_{m_i-2,m_i-1} & \psi^i_{m_i-1,m_i-1} & \psi^i_{m_i-1,m_i} \\ 0 & 0 & 0 & \cdots & 0 & \psi^i_{m_i-1,m_i} & \psi^i_{m_i,m_i} \end{bmatrix}. \tag{5.24}$$

Computing the terms from the diagonal of $\boldsymbol{\Psi}$. For the first and the last basis, i.e. for $j = 1$ and $j = m_i$, the integrals of the form $\psi^i_{j,j} = \int_0^1 \phi^2_{i,j}(x)dx$ correspond to the area of the square of right-angled triangles. Consider a right-angled triangle with height $h = 1$ and base b. Then, the area of the square of this triangle is given by,

$$\int_0^b \left[\frac{x}{b}\right]^2 dx = \frac{1}{b^2}\int_0^b x^2 dx = \frac{b}{3}. \tag{5.25}$$

In our case, we have values of b depending on the inter-spaces between knots. Then, using (5.25), we have

$$\psi^i_{1,1} = \int_0^1 \phi^2_{i,1}(x)dx = \frac{t_2 - t_1}{3}, \quad \text{and} \quad \psi^i_{m_i,m_i} = \int_0^1 \phi^2_{i,m_i}(x)dx = \frac{t_{m_i} - t_{m_i-1}}{3}. \tag{5.26}$$

For $j = 2, \cdots, m_i - 1$, one can note that $\psi^i_{j,j} = \int_0^1 \phi^2_{i,j}(x)dx$ correspond to the sum of the areas of the square of two right-angled triangles:

$$\psi^i_{j,j} = \int_{t_{j-1}}^{t_j} \phi^2_{i,j}(x)dx + \int_{t_j}^{t_{j+1}} \phi^2_{i,j}(x)dx = \frac{t_j - t_{j-1}}{3} + \frac{t_{j+1} - t_j}{3} = \frac{t_{j+1} - t_{j-1}}{3}. \tag{5.27}$$

Computing the terms outside the diagonal of $\boldsymbol{\Psi}$. We now aim at computing integrals of the form $\psi^i_{j,j+1} = \int_0^1 \phi_{i,j}(x)\phi_{i,j+1}(x)dx$, for $j = 1, \cdots, m-1$, for the upper bandwidth of (5.24). Since $\phi_{i,j}(x)\phi_{i,j+1}(x)$ has support $[t_j, t_{j+1}]$, then we have

$$\int_0^1 \phi_{i,j}(x)\phi_{i,j+1}(x)dx = \int_{t_j}^{t_{j+1}} \phi_{i,j}(x)\phi_{i,j+1}(x)dx.$$

This integral corresponds to the area of the product of two right-angled triangles where one of those is the "flipped" version of the other one w.r.t. the x-axis. Hence, we have to solve integrals of the form:

$$\int_0^b \left[\frac{x}{b}\right]\left[1 - \frac{x}{b}\right]dx = \frac{1}{b}\int_0^b \left[x - \frac{1}{b}x^2\right]dx = \frac{b}{6}. \tag{5.28}$$

Using (5.28), we obtain

$$\psi^i_{j,j+1} = \int_0^1 \phi_{i,j}(x)\phi_{i,j+1}(x)dx = \frac{t_{j+1} - t_j}{6}, \quad \text{for} \quad j = 1, \cdots, m_i - 1. \tag{5.29}$$

Then, (5.24) is computed using (5.26), (5.27), and (5.29), and Algorithm 2 can be used.

5.5.2 Proof in dimension d

5.5.2.A. Link between $\phi_{i,\cdot}$ and $\phi_{i+1,\cdot}$

Let $\boldsymbol{\beta}_i = [\mu^M_{i,1,\cdots,1}, \ \cdots, \mu^M_{i,m_{i,1},\cdots,m_{i,d}}] \in \mathbb{R}^{m_{i,1}\times\cdots\times m_{i,d}}$, and $\boldsymbol{\Phi}_i(\boldsymbol{x}) = \prod_{\kappa=1}^d \boldsymbol{\phi}_{i,\kappa}(x_\kappa)$ with

$$\boldsymbol{\phi}_{i,\kappa}(x_\kappa) = [\phi^\kappa_{i,1}(x_\kappa), \ \cdots, \ \phi^\kappa_{i,m_{i,\kappa}}(x_\kappa)]^\top.$$

Let the new hat basis functions $\boldsymbol{\Phi}_{i+1}(\boldsymbol{x}) = \prod_{\kappa=1}^d \boldsymbol{\phi}_{i+1,\kappa}(x_\kappa)$ with

$$\boldsymbol{\phi}_{i+1,\kappa}(x_\kappa) = [\phi^\kappa_{i+1,1}(x_\kappa), \ \cdots, \ \phi^\kappa_{i+1,m_{i+1,\kappa}}(x_\kappa)]^\top.$$

Here, we aim to express the inner product $\boldsymbol{\beta}_i^\top \boldsymbol{\Phi}_i(\boldsymbol{x})$ in terms of $\boldsymbol{\Phi}_{i+1}(\boldsymbol{x})$.

For instance, consider adding only one knot in the first dimension t_*^1 such that $t_\nu^1 < t_*^1 < t_{\nu+1}^1$ for $\nu = 1, \cdots, m_{i,1}-1$. Then, since the basis functions for the others dimensions remain equal, we have

$$\boldsymbol{\Phi}_i(\boldsymbol{x}) = \boldsymbol{\phi}_{i,1}(x_1) \times \left[\prod_{\kappa=2}^{d} \boldsymbol{\phi}_{i,\kappa}(x_\kappa)\right] = \boldsymbol{\phi}_{i,1}(x_1) \times \left[\prod_{\kappa=2}^{d} \boldsymbol{\phi}_{i+1,\kappa}(x_\kappa)\right].$$

Hence, we only need to express $\boldsymbol{\phi}_{i,1}$ in terms of $\boldsymbol{\phi}_{i+1,1}$. This can be done as is subsection 5.5.1, leading to expressions:

$$\phi_{i,j}^1(x_1) = \begin{cases} \phi_{i+1,j}^1(x_1), & \text{for } j \neq \nu, \nu+1, \\ \phi_{i+1,\nu}^1(x_1) + \phi_{i,\nu}^1(t_*^1)\phi_{i+1,\nu_1^*}^1(x_1), & \text{for } j = \nu, \\ \phi_{i+1,\nu+1}^1(x_1) + \phi_{i,\nu+1}^1(t_*^1)\phi_{i+1,\nu_1^*}^1(x_1), & \text{for } j = \nu+1, \end{cases} \tag{5.30}$$

where $\phi_{i,\nu}^1(t_*^1) = \frac{t_{\nu+1}-t_*}{t_{\nu+1}-t_\nu}$ and $\phi_{i,\nu+1}^1(t_*^1) = \frac{t_*-t_\nu}{t_{\nu+1}-t_\nu}$. Replacing (5.30) in (5.17), we have

$$\begin{aligned}
Y_{\mathcal{S}_i}^{\mathrm{MAP}}(\boldsymbol{x}) &= \sum_{j_1=1}^{m_{i,1}} \cdots \sum_{j_d=1}^{m_{i,d}} \mu_{i,j_1,\cdots,j_d}^M \prod_{\kappa=1}^{d} \phi_{i,j_\kappa}^\kappa(x_\kappa) \\
&= \sum_{\substack{j_1=1 \\ j_1 \neq \nu,\nu+1}}^{m_{i,1}} \cdots \sum_{j_d=1}^{m_{i,d}} \mu_{i,j_1,\cdots,j_d}^M \prod_{\kappa=1}^{d} \phi_{i,j_\kappa}^\kappa(x_\kappa) + \sum_{j_2=1}^{m_{i,2}} \cdots \sum_{j_d=1}^{m_{i,d}} \mu_{i,j_1=\nu,\cdots,j_d}^M \phi_{i,j_1=\nu}^1(x_1) \prod_{\kappa=2}^{d} \phi_{i,j_\kappa}^\kappa(x_\kappa) \\
&\quad + \sum_{j_2=1}^{m_{i,2}} \cdots \sum_{j_d=1}^{m_{i,d}} \mu_{i,j_1=\nu+1,\cdots,j_d}^M \phi_{i,j_1=\nu+1}^1(x_1) \prod_{\kappa=2}^{d} \phi_{i,j_\kappa}^\kappa(x_\kappa) \\
&= \sum_{\substack{j_1=1 \\ j_1 \neq \nu,\nu+1}}^{m_{i,1}} \cdots \sum_{j_d=1}^{m_{i,d}} \mu_{i,j_1,\cdots,j_d}^M \prod_{\kappa=1}^{d} \phi_{i+1,j_\kappa}^\kappa(x_\kappa) \\
&\quad + \sum_{j_2=1}^{m_{i,2}} \cdots \sum_{j_d=1}^{m_{i,d}} \mu_{i,j_1=\nu,\cdots,j_d}^M [\phi_{i+1,j_1=\nu}^1(x_1) + \phi_{i,j_1=\nu}^1(t_*^1)\phi_{i+1,\nu_1^*}^1(x_1)] \prod_{\kappa=2}^{d} \phi_{i+1,j_\kappa}^\kappa(x_\kappa) \\
&\quad + \sum_{j_2=1}^{m_{i,2}} \cdots \sum_{j_d=1}^{m_{i,d}} \mu_{i,j_1=\nu+1,\cdots,j_d}^M [\phi_{i+1,j_1=\nu+1}^1(x_1) + \phi_{i,j_1=\nu+1}^1(t_*^1)\phi_{i+1,\nu_1^*}^1(x_1)] \prod_{\kappa=2}^{d} \phi_{i+1,j_\kappa}^\kappa(x_\kappa) \\
&= \sum_{j_1=1}^{m_{i,1}} \cdots \sum_{j_d=1}^{m_{i,d}} \mu_{i,j_1,\cdots,j_d}^M \prod_{\kappa=1}^{d} \phi_{i+1,j_\kappa}^\kappa(x_\kappa) \\
&\quad + \sum_{j_2=1}^{m_{i,2}} \cdots \sum_{j_d=1}^{m_{i,d}} [\mu_{i,j_1=\nu,\cdots,j_d}^M \phi_{i,j_1=\nu}^1(t_*^1) + \mu_{i,j_1=\nu+1,\cdots,j_d}^M \phi_{i,j_1=\nu+1}^1(t_*^1)]\phi_{i+1,\nu_1^*}^1(x_1) \prod_{\kappa=2}^{d} \phi_{i+1,j_\kappa}^\kappa(x_\kappa) \\
&= {\boldsymbol{\beta}_i'}^\top \boldsymbol{\Phi}_{i+1}(\boldsymbol{x}),
\end{aligned}$$

with

$$\begin{aligned}
{\boldsymbol{\beta}_i'}^\top = [&\mu_{i,1,\cdots,1}^M, \cdots, \mu_{i,\nu,\cdots,1}^M \phi_{i,\nu}^1(t_*^1) + \mu_{i,\nu+1,\cdots,1}^M \phi_{i,\nu+1}^1(t_*^1), \cdots, \\
&\mu_{i,\nu,\cdots,m_{i,d}}^M \phi_{i,\nu}^1(t_*^1) + \mu_{i,\nu+1,\cdots,m_{i,d}}^M \phi_{i,\nu+1}^1(t_*^1), \cdots, \mu_{i,m_{i,1},\cdots m_{i,d}}^M].
\end{aligned} \tag{5.31}$$

5.5.2.A. Computing the matrix Ψ

Let $\boldsymbol{\Psi}_i$ be the matrix given by

$$\boldsymbol{\Psi}_{i+1} = \int_0^1 \cdots \int_0^1 \boldsymbol{\Phi}_{i+1}(\boldsymbol{x}) \boldsymbol{\Phi}_{i+1}^\top(\boldsymbol{x}) d\boldsymbol{x},$$

with $\boldsymbol{\Phi}_{i+1}(\boldsymbol{x}) = \prod_{\kappa=1}^d \boldsymbol{\phi}_{i+1,\kappa}(x_\kappa)$. Then, by using some properties on tensor products, one can show that

$$\begin{aligned}
\boldsymbol{\Psi}_{i+1} &= \int_0^1 \cdots \int_0^1 [\boldsymbol{\phi}_{i+1,1}(x_1) \times \cdots \times \boldsymbol{\phi}_{i+1,d}(x_d)][\boldsymbol{\phi}_{i+1,1}^\top(x_1) \times \cdots \times \boldsymbol{\phi}_{i+1,d}^\top(x_d)] dx_1 \cdots dx_d \\
&= \int_0^1 \boldsymbol{\phi}_{i+1,1}(x_1) \boldsymbol{\phi}_{i+1,1}^\top(x_1) dx_1 \times \cdots \times \int_0^1 \boldsymbol{\phi}_{i+1,d}(x_d) \boldsymbol{\phi}_{i+1,d}^\top(x_d) dx_d \\
&= \bigotimes_{\kappa=1}^d \widetilde{\boldsymbol{\Psi}}_{i+1,\kappa},
\end{aligned}$$

where $\widetilde{\boldsymbol{\Psi}}_{i+1,\kappa} = \int_0^1 \boldsymbol{\phi}_{i+1,\kappa}(x_\kappa) \boldsymbol{\phi}_{i+1,\kappa}^\top(x_\kappa) dx_\kappa$ has the same structure as the one obtained in subsection 5.2.2. Hence, (5.15) can be computed and (5.16) can be optimised.

Chapter 6

Additive Gaussian Processes under Inequality Constraints

Contents

6.1 Introduction

As discussed in Chapters 3 to 5, the main drawback of the tensor construction in (3.9) relies in its intractability in high dimensions. Since the number of terms in (3.9) increases exponentially with the dimension d, it results in costly (if not intractable) procedures for approximating the posterior of (3.9) via MC/MCMC algorithms. Experimental results in Chapters 4 and 5 have been restricted up to 5D examples and under certain computational simplifications (i.e. considering inactive inputs).

In this chapter, we adapt the framework of previous chapters to functions with an additive structure. We consider the additive GP Y in (2.10) satisfying inequality constraints everywhere. Since constraints are assumed to be imposed on a predefined subset of input variables due to additivity, developments proposed in Chapters 3 to 5 can be efficiently applied on (usually) low-dimensional subspaces. This leads to constrained (additive) GP models that can be easily scaled in dimensions involving hundreds of input variables.

Due to the additive form, the computations still involve truncated multinormals. However, a main difficulty is that the inequality constraints are not necessarily ensured everywhere if they are verified only at knots, in general. Favourable situations where this remains true are for monotonicity and convexity constraints. This results in a huge reduction of the complexity.

On various synthetic examples, we test the versatility and scalability of the proposed additive GP model to account for linear inequality constraints in high dimensions. We also assess its performance on the 5D coastal flooding application in subsection 4.3.2. Since MC and MCMC samplers are performed in low dimensions, numerical implementations are much faster compared to the ones proposed in Chapters 3 to 5. More precisely, while experimental results in previous chapters took tens of minutes, results here are obtained after a couple of seconds.

6.2 Finite-dimensional approximation of additive Gaussian processes

6.2.1 Approximation of additive Gaussian processes

Consider the centred additive GP $\{Y(\boldsymbol{x}); \boldsymbol{x} \in \mathcal{D}\}$ as in (2.10) with covariance function k in (2.12) and compact space $\mathcal{D} \in [0,1]^d$. Here, we aim at imposing some inequality constraints (e.g. boundedness, monotonicity, convexity) over Y. Consider Y_{p,m_p} for $p = 1, \cdots, d$, as the piecewise linear approximation of Y_p at knots $t_1^{(p)}, \cdots, t_{m_p}^{(p)}$:

$$Y_{p,m_p}(x_p) = \sum_{j_p=1}^{m_p} \xi_{j_p}^{(p)} \phi_{j_p}^{(p)}(x_p), \tag{6.1}$$

where $x_1, \cdots, x_d \in [0,1]$, $\xi_{j_p}^{(p)} := Y_p(t_{j_p}^{(p)})$, and $\phi_1^{(p)} \cdots, \phi_{m_p}^{(p)}$ are hat basis functions given by (3.2). Thus, the finite-dimensional approximation of (2.10) can be written as

$$Y_m(\boldsymbol{x}) = \sum_{p=1}^{d} \sum_{j_p=1}^{m_p} \xi_{j_p}^{(p)} \phi_{j_p}^{(p)}(x_p). \tag{6.2}$$

As in Chapter 3, we define $\boldsymbol{\xi}_p = [\xi_1^{(p)}, \cdots, \xi_{m_p}^{(p)}]^\top$, for $p = 1, \cdots, d$, as centred Gaussian vectors with covariance matrices $\boldsymbol{\Gamma}_p = (k_{p,m_p}(t_i^{(p)}, t_j^{(p)}))_{1 \leq i,j \leq m_p}$. We assume that the vectors $\boldsymbol{\xi}_p$'s are independent. Then, the covariance function in (2.11) can be written as:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{p=1}^{d} \left(\sum_{i_p=1}^{m_p} \sum_{j_p=1}^{m_p} \phi_{i_p}^{(p)}(x_p) k_{p,m_p}(t_{i_p}^{(p)}, t_{j_p}^{(p)}) \phi_{j_p}^{(p)}(x_p') \right), \tag{6.3}$$

with $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{D}$. The model in (6.2) can be extended to partially dependent sub-processes Y_{p,m_p} by adding additional cross-covariance terms in (6.3).

6.2.2 Conditioning to interpolation and inequality constraints

Consider the finite-dimensional representation of additive GPs as in (6.2). Following a similar procedure as in Chapter 3, the approximation Y_m given both interpolation and inequality constraints, can be written as

$$Y_m(\boldsymbol{x}) = \sum_{p=1}^{d} \sum_{j_p=1}^{m_p} \xi_{j_p}^{(p)} \phi_{j_p}^{(p)}(x_p) \quad \text{s.t.} \quad \begin{cases} Y_m(\boldsymbol{x}_i) + \varepsilon_i = y_i, \\ \boldsymbol{\xi}_p \in \mathcal{C}^{(p)}, \end{cases} \tag{6.4}$$

where $\boldsymbol{x}_i \in \mathcal{D}$, $y_i \in \mathbb{R}$ for $i = 1, \cdots, n$, and $\mathcal{C}^{(p)}$ a convex set as in (3.10). As in Chapter 4, we consider an additive Gaussian noise $\varepsilon_i \sim \mathcal{N}(0, \tau^2)$ with noise variance τ^2, and we assume that $\varepsilon_1, \cdots, \varepsilon_n$ are independent, and independent of Y_m. Given a DoE $\mathbf{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]^\top$, we have matricially:

$$\mathbf{Y}_m = \left[Y_m(\boldsymbol{x}_1), \quad \cdots, \quad Y_m(\boldsymbol{x}_n)\right]^\top = \sum_{p=1}^{d} \boldsymbol{\Phi}_p \boldsymbol{\xi}_p,$$

with $\boldsymbol{\Phi}_p$ an $n \times m_p$ matrix defined by $(\boldsymbol{\Phi}_p)_{i,j} = \phi_{j_p}^{(p)}(x_p^i)$. Denote $\boldsymbol{\Lambda}_p = (\lambda_{i,j}^{(p)})_{1 \leq i \leq q, 1 \leq j \leq m}$, $\boldsymbol{l}_p = (l_i^{(p)})_{1 \leq i \leq q}$, and $\boldsymbol{u}_p = (u_i^{(p)})_{1 \leq i \leq q}$ the set of linear inequality conditions of $\boldsymbol{\xi}_p$. Then, the distribution of $\boldsymbol{\xi}_p$, for $p = 1, \cdots, d$, given both interpolation and inequality conditions is truncated multinormal:

$$\boldsymbol{\xi}_p \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}_p) \, \forall \, p = 1, \cdots, d \quad \text{s.t.} \quad \begin{cases} \sum_{p=1}^{d} \boldsymbol{\Phi}_p \boldsymbol{\xi}_p + \tau^2 \mathbf{I} = \boldsymbol{y} \\ \boldsymbol{l}_p \leq \boldsymbol{\Lambda}_p \boldsymbol{\xi}_p \leq \boldsymbol{u}_p. \end{cases} \tag{6.5}$$

Finally, as shown in subsection 3.3.1, and using (2.13), the posterior distribution of (6.5) is obtained from

$$\boldsymbol{\Lambda}_p \boldsymbol{\xi}_p \left| \left\{ \sum_{p=1}^{d} \boldsymbol{\Phi}_p \boldsymbol{\xi}_p + \tau^2 \mathbf{I} = \boldsymbol{y}, \boldsymbol{l}_p \leq \boldsymbol{\Lambda}_p \boldsymbol{\xi}_p \leq \boldsymbol{u}_p \right\} \sim \mathcal{TN}\left(\boldsymbol{\Lambda}_p \boldsymbol{\mu}_p, \; \boldsymbol{\Lambda}_p \boldsymbol{\Sigma}_p \boldsymbol{\Lambda}_p^\top, \; \boldsymbol{l}_p, \; \boldsymbol{u}_p\right), \right. \tag{6.6}$$

where

$$\boldsymbol{\mu}_p = \boldsymbol{\Gamma}_p \boldsymbol{\Phi}_p^\top \mathbf{C}^{-1} \boldsymbol{y}, \quad \boldsymbol{\Sigma}_p = \boldsymbol{\Gamma}_p - \boldsymbol{\Gamma}_p \boldsymbol{\Phi}_p^\top \mathbf{C}^{-1} \boldsymbol{\Phi}_p \boldsymbol{\Gamma}_p, \tag{6.7}$$

with $\mathbf{C} = \sum_{p=1}^{d} \boldsymbol{\Phi}_p \boldsymbol{\Gamma}_p \boldsymbol{\Phi}_p^\top + \tau^2 \mathbf{I}$. Note that the computation of $\mathbf{C}^{-1}$ can be considerable speeded-up using the matrix inversion lemma (Press et al., 1992; Rasmussen and Williams, 2005). Consider the extended matrix of basis functions $\boldsymbol{\Psi} = [\boldsymbol{\Phi}_1, \cdots, \boldsymbol{\Phi}_d]^\top$ and the block diagonal matrix $\boldsymbol{\Upsilon} = \text{bdiag}(\boldsymbol{\Gamma}_1, \cdots, \boldsymbol{\Gamma}_d)$. Thus, using the matrix inversion lemma (see, e.g., Rasmussen and Williams, 2005, appendix A.3), we have

$$\mathbf{C}^{-1} = (\boldsymbol{\Psi} \boldsymbol{\Upsilon} \boldsymbol{\Psi}^\top + \tau^2 \mathbf{I})^{-1} = \tau^{-2} \mathbf{I} - \tau^{-4} \boldsymbol{\Psi} (\boldsymbol{\Upsilon}^{-1} + \tau^{-2} \boldsymbol{\Psi}^\top \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top,$$

with $\boldsymbol{\Upsilon}^{-1} = \text{bdiag}(\boldsymbol{\Gamma}_1^{-1}, \cdots, \boldsymbol{\Gamma}_d^{-1})$. Furthermore, a similar equation exists for the determinant of $\mathbf{C}$:

$$\det(\mathbf{C}) = \det(\boldsymbol{\Psi}\boldsymbol{\Upsilon}\boldsymbol{\Psi}^\top + \tau^2\mathbf{I}) = \tau^{2n}\det(\boldsymbol{\Upsilon})\det(\boldsymbol{\Upsilon}^{-1} + \tau^{-2}\boldsymbol{\Psi}^\top\boldsymbol{\Psi}),$$

with $\det(\boldsymbol{\Upsilon}) = \det(\boldsymbol{\Gamma}_1) \times \cdots \times \det(\boldsymbol{\Gamma}_d)$. The computation of $\det(\mathbf{C})$ is required for the covariance parameter estimation via ML. Observe that, for $m = m_1 \times \cdots \times m_d \ll n$, both $\mathbf{C}^{-1}$ and $\det(\mathbf{C})$ can be efficiently computed since (highly parallelizable) inversion and determinant operations are applied over $m \times m$ full-rank matrices.

Finally, (6.6) can be approximated via MC/MCMC. Denoting $\boldsymbol{\eta}_p = \boldsymbol{\Lambda}_p\boldsymbol{\xi}_p$, notice that samples for $\boldsymbol{\xi}_p$ can be obtained by sampling from $\boldsymbol{\eta}_p$ and solving a linear system (we refer to Chapter 3 for a further discussion). Furthermore, the MAP estimates $\boldsymbol{\mu}_1^*, \cdots, \boldsymbol{\mu}_d^*$ are obtained by solving d quadratic optimisation problems as in (3.15) using the conditional parameters in (6.7). Observe also that, since the inequality constraints are imposed in 1D sub-processes Y_{p,m_p}, for $p = 1, \cdots, d$, MC and MCMC samplers can be efficiently performed as shown in Chapter 3. Furthermore, the approximation of the posteriors of the d sub-models in (6.6) and the corresponding MAP estimates can be parallelised in order to reduce the computational cost (i.e. execution time).

Hence, as we have just shown, computations with additive GPs under inequality constraints are done in the same way as in Chapter 3 and involve truncated multinormals. A new issue is that certain component-wise inequality conditions over the sub-processes Y_{p,m_p} do not imply that they will be globally satisfied everywhere. Intuitively, considering a function of class $f \in C^\infty$ such that $f(\boldsymbol{x}) = f_1(x_1) + \cdots + f_d(x_d)$. One may note that any inequality conditions over the derivatives of f holds for $f_1, \cdots, f_d$. For example, for non-decreasing constraints, we have that $\frac{\partial}{\partial \boldsymbol{x}} f(\boldsymbol{x}) = [\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_d}]^\top = [\frac{\partial f_1}{\partial x_1}, \cdots, \frac{\partial f_d}{\partial x_d}]^\top$. Thus, $\frac{\partial}{\partial \boldsymbol{x}} f(\boldsymbol{x}) \geq \mathbf{0} \Leftrightarrow \frac{\partial}{\partial x_p} f(x_p) \geq 0$ for $p = 1, \cdots, d$. A similar analysis can be done for convexity but considering the Hessian of f:

$$H_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial^2 f}{\partial x_d^2} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 f_1}{\partial x_1^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial^2 f_d}{\partial x_d^2} \end{bmatrix}$$

Then, having p.s.d. condition on H_f implies that $\frac{\partial^2}{\partial x_p^2} f(\boldsymbol{x}) \geq 0 \Leftrightarrow \frac{\partial^2}{\partial x_p^2} f(x_p) \geq 0$ for $p = 1, \cdots, d$.

In the next propositions, we study the cases where either boundedness, monotonicity or convexity constraints are ensured everywhere on the finite-dimensional GP Y_m.

Proposition 6.1 *Consider a finite-dimensional additive (centred) GP* $Y_m(x_1, \cdots, x_d) = Y_{1,m_1}(x_1) + \cdots + Y_{d,m_d}(x_d)$. *Then,*

$$Y_m \in [l, u] \quad \not\Leftrightarrow \quad Y_{p,m_p} \in [l, u] \ \forall\ p = 1, \cdots, d,$$

for $-\infty < l < u < \infty$

Proof. Without loss of generality, we consider a 2D additive (centred) GP $Y_m(x_1, x_2) = Y_{1,m_1}(x_1) + Y_{2,m_2}(x_2)$.

$\Rightarrow$ Assume that $Y_m \in [l, u = 2l]$ for $0 < l < \infty$. Then, by definition, we have

$$l \leq Y_{1,m_1} + Y_{2,m_2} \leq 2l.$$

Assume now that $Y_{2,m_2} \in [l, 2l]$. For the case when $Y_{2,m_2} = l$, we have that $Y_{1,m_1} \in [0, l]$. Hence, for $0 \leq Y_{1,m_1} < l$, we deduce that the equivalence $Y_m \in [l, 2l] \;\Rightarrow\; Y_{p,m_p} \in [l, 2l]$, for $p = 1, 2$, does not holds.

$\Leftarrow$ Assume that $Y_{p,m_p} \in [l, u]$ for $p = 1, 2$. Then, for the case $Y_{1,m_1} = Y_{2,m_2} = u$, we have that $Y_m = 2u$, for $u > 0$, and hence $Y_m \notin [l, u]$. □

For the case of globally accounting for $Y_m \in [l, u]$, we need to impose those constraints as in Chapter 3.

Proposition 6.2 *Consider the additive GP in Proposition 6.1. Then,*

$$Y_m \in \mathcal{E}_{1,d} \quad \Leftrightarrow \quad Y_{p,m_p} \in \mathcal{E}_1 \;\forall\; p = 1, \cdots, d,$$

where $\mathcal{E}_{1,d}$ and $\mathcal{E}_1$ are the convex sets of functions satisfying monotonicity conditions in $\mathbb{R}^d$ and $\mathbb{R}$, respectively.

Proof. As in Proposition 6.1, we assume a 2D additive GP $Y_m(x_1, x_2) = Y_{1,m_1}(x_1) + Y_{2,m_2}(x_2)$. Let (x_1, x_2) and (x_1', x_2') be two points such that $x_1 \leq x_1'$ and $x_2 \leq x_2'$.

$\Rightarrow$ Assume that $Y_m \in \mathcal{E}_{1,d}$. By definition, we have

$$Y_m(x_1, x_2) = Y_{1,m_1}(x_1) + Y_{2,m_2}(x_2) \leq Y_{1,m_1}(x_1') + Y_{2,m_2}(x_2') = Y_m(x_1', x_2').$$

Then, assuming that $x_1 < x_1'$ and $x_2 = x_2'$, we deduce that $Y_{1,m_1}(x_1) \leq Y_{1,m_1}(x_1')$. The same conclusion can be reached when $x_1 = x_1'$ and $x_2 < x_2'$.

$\Leftarrow$ Assume that $Y_{p,m_p} \in \mathcal{E}_1$ for $p = 1, 2$. This implies that $Y_{1,m_1}(x_1) \leq Y_{1,m_1}(x_1')$ and $Y_{2,m_2}(x_2) \leq Y_{2,m_2}(x_2')$. Hence, $Y_{1,m_1}(x_1) + Y_{2,m_2}(x_2) \leq Y_{1,m_1}(x_1') + Y_{2,m_2}(x_2')$, and the proposition holds. □

Proposition 6.3 *Consider the additive GP in Proposition 6.1. Then,*

$$Y_m \in \mathcal{E}_{2,d} \quad \Leftrightarrow \quad Y_{p,m_p} \in \mathcal{E}_2 \;\forall\; p = 1, \cdots, d,$$

where $\mathcal{E}_{2,d}$ and $\mathcal{E}_2$ are the sets of functions satisfying convexity conditions in $\mathbb{R}^d$ and $\mathbb{R}$, respectively.

Proof. As in Proposition 6.1, we assume a 2D additive GP $Y_m(x_1, x_2)$. Let us first define the convexity in $\mathbb{R}^2$. Remind that Y_m is called convex if and only if:

$$Y_m(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{x}') \leq \lambda Y_m(\boldsymbol{x}) + (1-\lambda) Y_m(\boldsymbol{x}'), \quad \forall\; \lambda \in [0, 1], \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^2. \tag{6.8}$$

$\Rightarrow$ Assume that $Y_m \in \mathcal{E}_{2,d}$. For a fixed value $x_2 = a$, for $a \in \mathbb{R}$, we have

$$Y_{1,m_1}(x_1) = Y_m(x_1, a) - Y_{2,m_2}(a).$$

Then, since $Y_m \in \mathcal{E}_{2,d}$, we have that $Y_m(x_1, a)$ satisfies,

$$Y_m\left(\lambda \begin{bmatrix} x_1 \\ a \end{bmatrix} + (1-\lambda) \begin{bmatrix} x_1' \\ a \end{bmatrix}\right) \leq \lambda Y_m\left(\begin{bmatrix} x_1 \\ a \end{bmatrix}\right) + (1-\lambda) Y_m\left(\begin{bmatrix} x_1' \\ a \end{bmatrix}\right).$$

Thus, since $Y_m(x_1, a) \in \mathcal{E}_2$, with $Y_{2,m_2}(a)$ being constant, we deduce that $Y_{1,m_1}(x_1) \in \mathcal{E}_2$. The same conclusion can be reached considering $x_1 = a$ and x_2.

$\Leftarrow$ Assume that $Y_{p,m_p} \in \mathcal{E}_2$ for $p = 1, 2$. By the definition of convexity, we have

$$\begin{aligned} Y_{1,m_1}(\lambda x_1 + (1-\lambda)x_1') &\leq \lambda Y_{1,m_1}(x_1) + (1-\lambda) Y_{1,m_1}(x_1'), \\ Y_{2,m_2}(\lambda x_2 + (1-\lambda)x_2') &\leq \lambda Y_{2,m_2}(x_2) + (1-\lambda) Y_{2,m_2}(x_2'), \end{aligned}$$

for $\lambda \in [0, 1]$. Then, since $Y_m(x_1, x_2) = Y_{1,m_1}(x_1) + Y_{2,m_2}(x_2)$, condition in (6.8) holds. $\square$

6.2.3 Numerical illustrations

We now show on synthetic examples that the framework enables us to address different types of constraints. For instance, the value of m is manually fixed aiming a high quality of resolution along each dimension. However, one must note that the sequential algorithm suggested in Chapter 5 can be combined in order to reduce the computational cost.

6.2.3.1 2D toy example under monotonicity and convexity constraints

We continue with the toy example in Figure 2.7 where interpolation conditions have been obtained from function $(x_1, x_2) \mapsto 4(x_1 - 0.5)^2 + 2x_2$ at points $(x_i, y_i)_{1 \leq i \leq 5}$: $(0.5, 0)$, $(0.5, 0.5)$, $(0.5, 1)$, $(0, 0.5)$, and $(1, 0.5)$. For this example, the target function satisfies component-wise convexity and non-decreasing constraints along the first and second dimension, respectively. From Figure 2.7, one could observe that these conditions were not fulfilled by the unconstrained GP. We now suggest an additive GP model accounting for both constraints and with the same parametrisation as the one used to obtain Figure 2.7.[1]

From Figure 6.1, one can observe that the quality of predictions are significantly improved by enforcing the model to both monotonicity and convexity constraints. More precisely, we obtain an absolute improvement about 9% of the Q^2 indicator, resulting in a $Q^2 = 98.5\%$ (compared to the 89.6% obtained in Figure 2.7).

[1] Consider an additive function $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$. Then, similarly to Propositions 6.2 and 6.3, it can be shown that f is convex w.r.t. x_1 and non-decreasing w.r.t. x_2 if and only if f_1 is convex and f_2 is non-decreasing.

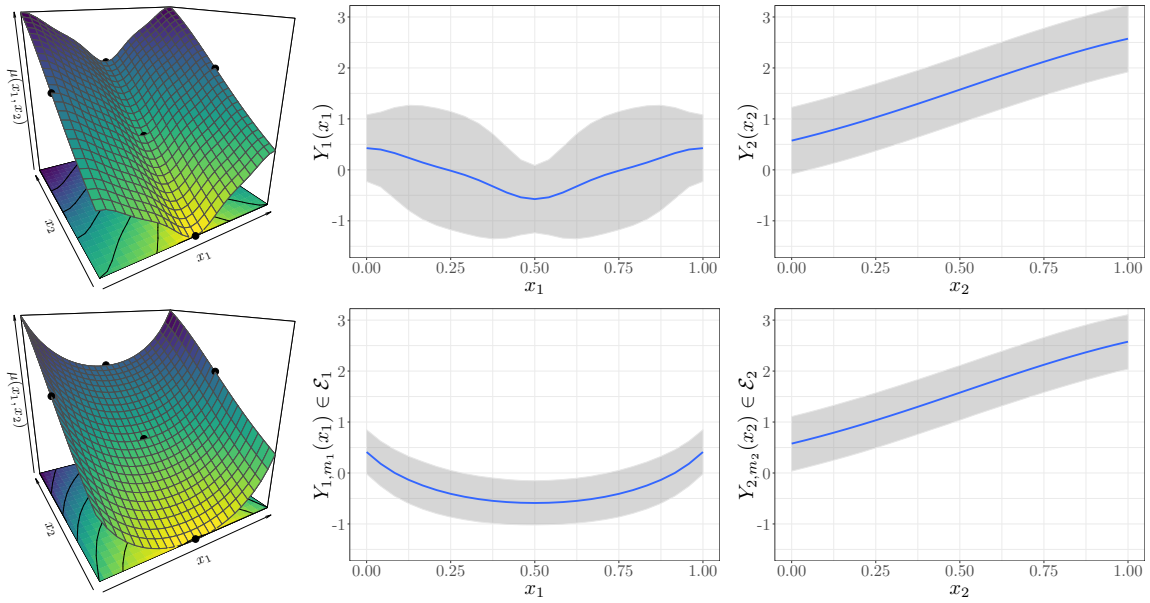


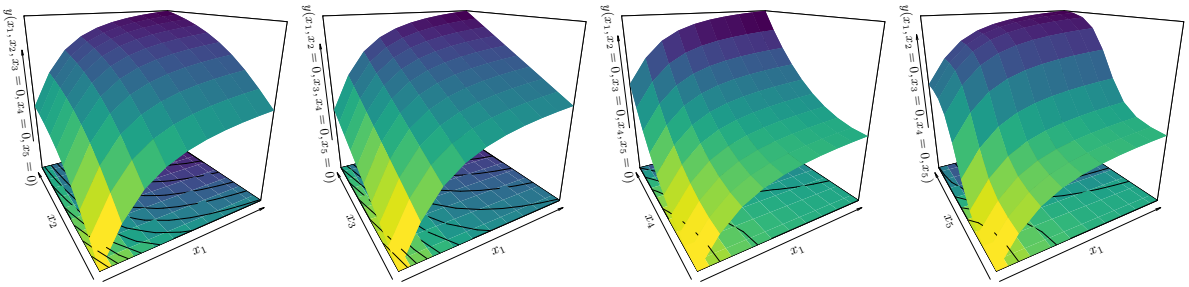
Figure 6.1: Conditional additive GPs under inequality constraints in 2D. Results are shown (top) without constraints and (bottom) with component-wise convexity and monotonicity constraints. Each row shows: the conditional mean function $\mu(x_1, x_2) = \mu_1(x_1) + \mu_2(x_2)$ (left), and the sub-models Y_{1,m_1} (centre) and Y_{2,m_2} (right). For $\mu(x_1, x_2)$, black dots represent the interpolation points. For the sub-models, the mean functions μ_1, μ_2 (blue solid lines) and 90% of the conditional interval (grey region) are shown.

6.2.3.2 5D toy example under monotonicity constraints

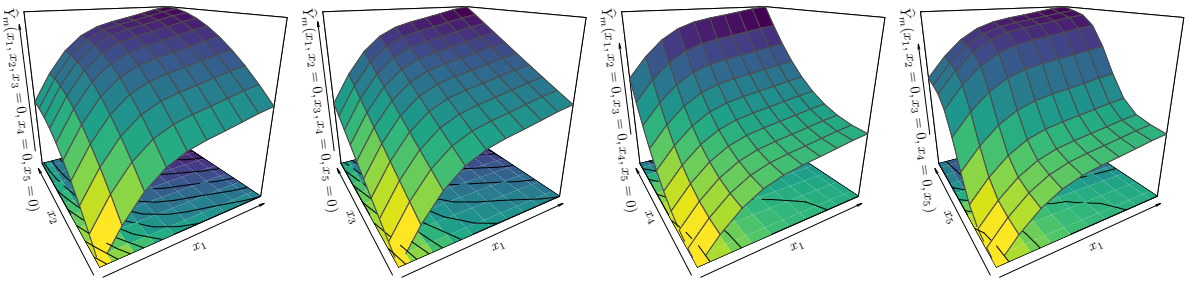
We continue with the 5D toy example of Figure 4.5 under monotonicity constraints. There, noisy evaluations of (4.6) on a maximin Latin-hypercube DoE over $[0, 1]^5$ at 2000 locations have been taken. Here, we exploit the fact that the function in (4.6) is additive.

We use an additive 5D SE covariance function and estimate the parameters $\boldsymbol{\theta} = ((\sigma_1^2, \ell_1), \cdots, (\sigma_5^2, \ell_5), \tau)$ via ML. Note that, contrarily to the (non-additive) GP model suggested in subsection 4.2.3, we now have five variance parameters $\sigma_1^2, \cdots, \sigma_5^2$ to estimate (instead of the single one used there). Since the additive GP model scales better w.r.t. the number of dimensions due to its non-tensorised structure, here we are able to increase the quality of resolution in the approximation. Thus, we fixed the number of knots per dimension to twenty, i.e. $m_1 = \cdots = m_5 = 20$.

Figure 6.2 shows the predictions obtained by the additive GP model under monotonicity conditions. As in subsection 4.2.3, 10^4 monotonic trajectories were sampled via HMC in order to approximate the posterior distribution. In contrast to the CPU running time of 44.8 minutes obtained there, samples here were obtained after a lapse of 5 seconds on a single core of the same machine. One must note that faster sampling implementations could be obtained by computing the sub-model $Y_{1,m_1}, \cdots, Y_{5,m_5}$ as independent processes on a multiple core processor.



(a) $y(x_1, x_2, 0, 0, 0)$ (b) $y(x_1, 0, x_3, 0, 0)$ (c) $y(x_1, 0, 0, x_4, 0)$ (d) $y(x_1, 0, 0, 0, x_5)$



(e) $\widehat{Y}_m(x_1, x_2, 0, 0, 0)$ (f) $\widehat{Y}_m(x_1, 0, x_3, 0, 0)$ (g) $\widehat{Y}_m(x_1, 0, 0, x_4, 0)$ (h) $\widehat{Y}_m(x_1, 0, 0, 0, x_5)$

Figure 6.2: Additive GP model under monotonicity constraints in 5D. True and predictive mean profiles are shown in the first and second row, respectively.

6.2.3.3 Monotonicity in hundreds of dimensions

In order to test the proposed framework in high dimensions, we consider the additive function:

$$y(\boldsymbol{x}) = \sum_{p=1}^{d} \arctan\left(5\left[1 - \frac{p}{d}\right] x_p\right), \tag{6.9}$$

with $\boldsymbol{x} = (x_1, \cdots, x_d) \in [0,1]^d$. Note that (6.9) is completely monotone with different growth rates along each dimension. Small values of p leads to high growth rates.

We train GP models under monotonicity constraints considering various values of d. Then, (6.9) is evaluated at $10 \times d$ locations on a maximim Latin-hypercube DoE over $[0,1]^d$. The number of knots $m_1 = \cdots = m_d = 5$ are set aiming a trade-off between high-quality of approximation and computational time. We use additive SE kernels with fixed covariance parameters $(\sigma_p^2, \ell_p) = (1, 2)$ for $p = 1, \cdots, d$. Those parameters can be estimated via ML but here we aim at testing the computational cost of both prediction and HMC sampling steps.

Running times for both the prediction and 10^3 HMC samples are shown in Table 6.1. We observe that, while simulations are fast, the main cost relies in the prediction step. One must note that the latter step involves the computation of d sets of inequality conditions, conditional parameters in (6.7) and d MAP estimates. However, as discussed

Table 6.1: Computational cost of predictions and HMC samples of an additive GP under monotonicity constraints in dimension d. CPU times are computed for sampling 10^3 monotonic trajectories using the HMC sampler from (Pakman and Paninski, 2014).

CPU Time $[s]$	d								
	2	5	10	20	50	100	200	500	1000
Prediction	0.01	0.01	0.02	0.03	0.25	1.37	10.48	165.85	1364.54
HMC Sampling	0.05	0.11	0.22	0.72	1.14	2.91	2.73	5.28	10.83

in subsection 6.2.3.2, those calculations can be parallelised for faster implementations. Finally, predictions in 10^3 dimensions are shown in Figure 6.3. One can observe that the constrained GP model is able to capture the monotonic behaviour of (6.9).

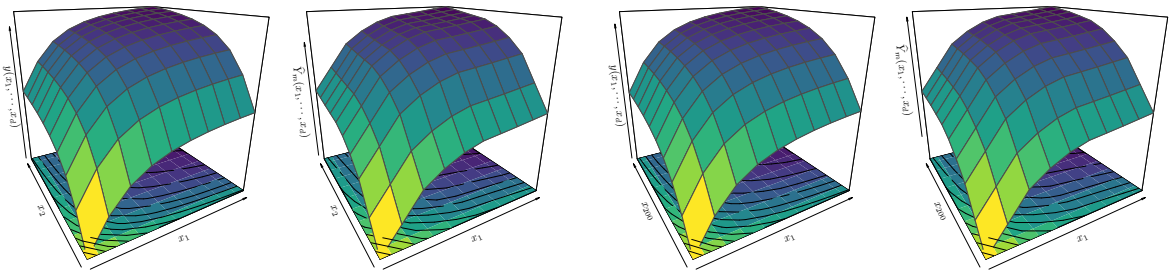
6.2.3.4 5D coastal flooding application

We continue with the 5D coastal flooding example in subsection 4.3.2. Here, we assess the performance of an additive GP model under monotonicity conditions along the first and the second input dimensions. We use the same parametrisation proposed in subsection 4.3.2. We normalise the input space to be in $[0,1]^5$, and use a 5D additive Matérn 5/2 kernel. The parameters $((\sigma_1^2,\ell_1),\cdots,\cdots,(\sigma_5^2,\ell_5),\tau^2)$ are estimated via ML. As the additive construction in (6.4) allows us to have a higher resolution of representation compared to the tensor approach in Chapter 4, here we fix twenty knots per dimension.

As in subsection 4.3.2, we trained GP models using twenty different sets of training data and different proportions of training sets. According to Figure 6.4, one can observe that the additive model could not properly learn the behaviour of data. This results from the fact that the 5D coastal data is not fully additive. However, one must note that numerical experiments here were much more faster: in the order of seconds compared to the dozens of minutes required in subsection 4.3.2.

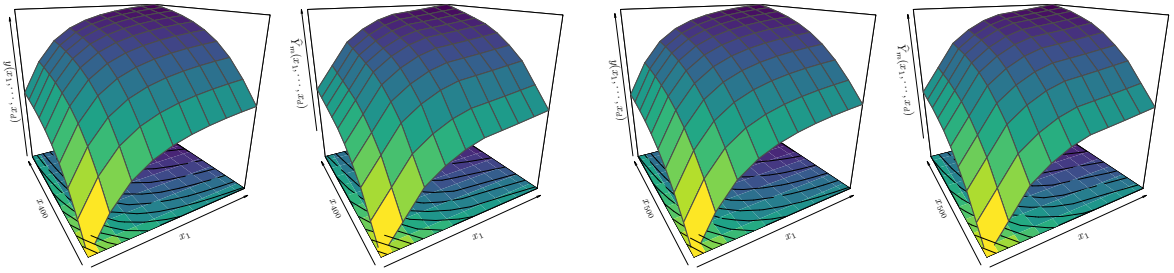
6.3 Future works: block additivity

As shown in subsection 6.2.3, the finite-dimensional GP in (6.4) leads to accurate predictions if additive functions are considered. However, the main drawback of the model in (6.1) is that it does not take into account interactions between input variables. Real-world applications satisfying the additive structure in (6.1) are not common but a wide range exhibits block-additivity (see, e.g., Duvenaud et al., 2011; Macdonald, 1998). In this section, we give an insight to extend the approximation in (6.4) to account for block-additive functions.



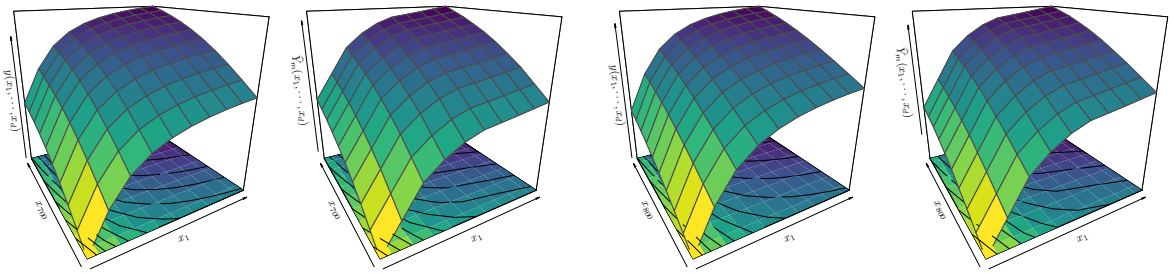
(a) $y_1(x_1) + y_2(x_2)$

(b) $y_1(x_1) + y_{200}(x_{200})$



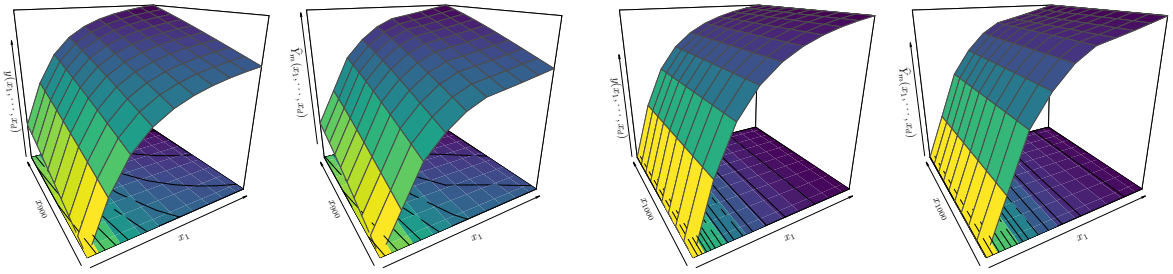
(c) $y_1(x_1) + y_{400}(x_{400})$

(d) $y_1(x_1) + y_{500}(x_{500})$



(e) $y_1(x_1) + y_{700}(x_{700})$

(f) $y_1(x_1) + y_{800}(x_{800})$



(g) $y_1(x_1) + y_{900}(x_{900})$

(h) $y_1(x_1) + y_{1000}(x_{1000})$

Figure 6.3: Examples of an additive GP under monotonicity constraints in 10^3 dimensions. Each panel shows: the true (left) and predictive (right) mean profiles.

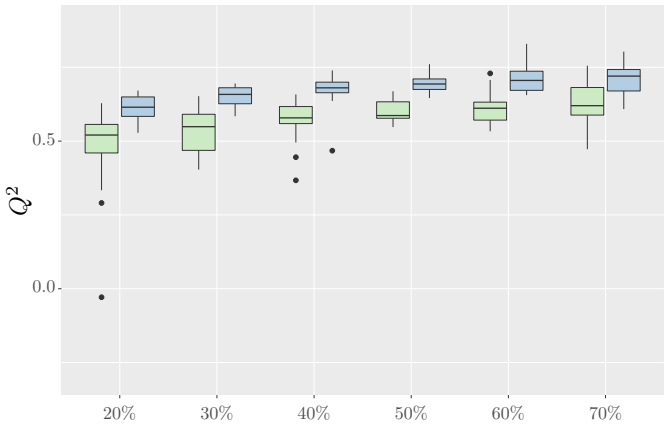


Figure 6.4: 5D additive GP emulators for modelling the coastal flooding data in (Azzimonti et al., 2019). The boxplots show the Q^2 results using different proportions of training points (x-axis) and using twenty different random training sets. Results are shown for the non-additive (blue) and additive (green) constrained GP emulators.

6.3.1 Additivity per blocks in 3D

Without lost of generality, consider an additive GP $\{Y(\boldsymbol{x}); \boldsymbol{x} \in [0,1]^3\}$,

$$Y(\boldsymbol{x}) = Y_1(x_1) + Y_{2,3}(x_2, x_3), \tag{6.10}$$

with sub-models $Y_1, Y_{2,3}$. Here, we assume that Y_1 and $Y_{2,3}$ are independent centred GPs with covariance functions k_1 and $k_{1,2}$. Since the cross-covariance function between Y_1 and $Y_{2,3}$ is equal to zero, the covariance function k of the GP in (6.10) is given by

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_1(x_1, x_1') + k_{2,3}(x_2, x_3; x_2', x_3'). \tag{6.11}$$

We aim at imposing constraints over the partitions $Y_1, Y_{2,3}$. Define a finite-dimensional GP, denoted by Y_{1,m_1}, as the piecewise linear interpolation of Y_1 at knots $(t_1^{(1)}, \cdots, t_{m_1}^{(1)})$ as in (3.1). Define also a 2D finite-dimensional GP, denoted by $Y_{(2,3),m_{2,3}}$, as the piecewise linear interpolation of $Y_{2,3}$ at a set of knots per dimension $(t_1^{(2)}, \cdots, t_{m_2}^{(2)}), \cdots, (t_1^{(3)}, \cdots, t_{m_3}^{(3)})$ as in (3.16). Thus, the finite-dimensional GP of (6.10) is given by

$$Y_m(\boldsymbol{x}) = \sum_{j_1=1}^{m_1} \phi_{j_1}^{(1)}(x_1)\xi_{j_1}^{(1)} + \sum_{j_2=1}^{m_2}\sum_{j_3=1}^{m_3} \phi_{j_2}^{(2)}(x_2)\phi_{j_3}^{(3)}(x_3)\xi_{j_2,j_3}^{(2,3)}, \tag{6.12}$$

where $\xi_{j_1}^{(1)} := Y_1(t_{j_1}^{(1)})$, $\xi_{j_2,j_3}^{(2,3)} := Y_{2,3}(t_{j_2}^{(2)}, t_{j_3}^{(3)})$ and $\phi_{j_1}^{(1)}, \phi_{j_2}^{(2)}, \phi_{j_3}^{(3)}$ are hat basis functions as in (3.2). Consider the centred Gaussian vectors $\boldsymbol{\xi}_1 = [\xi_1^{(1)}, \cdots, \xi_{m_1}^{(1)}]^\top$, with covariance matrix $\boldsymbol{\Gamma}_1 = (k_{1,m_1}(t_i^{(1)}, t_j^{(1)}))_{1\le i,j\le m_1}$, and $\boldsymbol{\xi}_{2,3} = [\xi_{1,1}^{(2,3)}, \cdots, \xi_{1,m_3}^{(2,3)}, \cdots, \xi_{m_2,1}^{(2,3)}, \cdots, \xi_{m_2,m_3}^{(2,3)}]^\top$, with covariance matrix $\boldsymbol{\Gamma}_{2,3} = (k_{(2,3),m_{2,3}}(t_i^{(2)}, t_j^{(3)}; t_{i'}^{(2)}, t_{j'}^{(3)}))_{1\le i,i'\le m_2, 1\le j,j'\le m_3}$. Assume that vectors $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_{2,3}$ are independent. Thus, the covariance function of (6.12) is given by

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_1(x_1, x_1') + k_{2,3}(x_2, x_3; x_2', x_3'), \tag{6.13}$$

with

$$k_1(x_1, x_1') = \sum_{i=1}^{m_1}\sum_{j=1}^{m_1} \phi_i^{(1)}(x_1) k_{1,m_1}(t_i^{(1)}, t_j^{(1)}) \phi_j^{(1)}(x_1'),$$

$$k_{2,3}(x_2, x_3; x_2', x_3') = \sum_{i,i'=1}^{m_2}\sum_{j,j'=1}^{m_3} \phi_i^{(2)}(x_2)\phi_j^{(3)}(x_3) k_{(2,3),m_{2,3}}(t_i^{(2)}, t_j^{(3)}; t_{i'}^{(2)}, t_{j'}^{(3)}) \phi_{i'}^{(2)}(x_2')\phi_{j'}^{(3)}(x_3').$$

Following a similar procedure as the one in Chapter 3, then the approximation Y_m given both interpolation and inequality constraints is given by

$$Y_m(\boldsymbol{x}) = \sum_{j_1=1}^{m_1} \phi_{j_1}^{(1)}(x_1)\xi_{j_1}^{(1)} + \sum_{j_2=1}^{m_2}\sum_{j_3=1}^{m_3} \phi_{j_2}^{(2)}(x_2)\phi_{j_3}^{(3)}(x_3)\xi_{j_2,j_3}^{(2,3)} \quad \text{s.t.} \quad \begin{cases} Y_m(\boldsymbol{x}_i) + \varepsilon_i = y_i, \\ \boldsymbol{\xi}_1 \in \mathcal{C}^{(1)}, \\ \boldsymbol{\xi}_{2,3} \in \mathcal{C}^{(2,3)}, \end{cases}$$

with $\boldsymbol{x}_i = (x_1^i, x_2^i, x_3^i) \in [0,1]^3$, $y_i \in \mathbb{R}$ for $i = 1, \cdots, n$, and convex sets $\mathcal{C}^{(1)}, \mathcal{C}^{(2,3)}$ defined in (3.3). We consider an additive Gaussian noise $\varepsilon_i \sim \mathcal{N}(0, \tau^2)$ with noise variance τ^2, and assume that $\varepsilon_1, \cdots, \varepsilon_n$ are independent, and independent of Y_m. Note that, since constraints are enforced up to two-dimensional spaces, the HMC sampler in (Pakman and Paninski, 2014) can be efficiently applied as shown in Chapters 3 to 5. Furthermore, sub-models Y_{1,m_1} and $Y_{(2,3),m_{2,3}}$ can be computed in parallel in order to reduce the execution time.

6.3.2 Remarks on implementations in high dimensions

Developments in subsection 6.3.1 can be generalised to $d > 3$ but at the cost of more cumbersome expressions. In that case, one should rewrite first (6.10) according to the block-additive structure of Y. Then, one can follow a similar procedure to the one previously discussed, and enforce the inequality constraints over each group as discussed throughout Chapters 3 to 5. Note that the computational complexity of the block-additive model will depend on the sizes of the blocks.

In subsection 6.3.1, we assumed that the additive structure of Y was known. In practice, this is not always the case and that structure has to be properly inferred. In that case, dedicate statistical approaches can be coupled here in order to define first the additive partitions (see, e.g., Fruth et al., 2014; Muehlenstaedt et al., 2012; Pan and Zhu, 2017), and then the procedure in subsection 6.3.1 can be established.

6.4 Conclusions

In this chapter, we adapted the GP model with inequality constraints to additive functions. Since constraints are assumed to be imposed on a (predefined) subset of input variables, MC/MCMC samplers in subsection 3.3.2 are then applied on one-dimensional subspaces. This leads to models that can be easily scaled in high dimensions involving

tens or hundreds of input variables. Furthermore, many of the computations can be efficiently computed and are highly parallelizable. However, a main difficulty is that the inequality constraints are not necessarily ensured everywhere if there are verified only at knots, in general. Favourable situations where this remains true are for monotonicity and convexity constraints. Finally, we also gave an insight of its extension to account for block-additive functions. In that case, enforcing the constraints over each block can be considered as in Chapters 3 to 5. Although this extension can be theoretically formalised for an arbitrary dimension d and arbitrary sets of additive blocks, its practical and general implementations is challenging. This could be of interest in future work.

The proposed additive GP model was tested on both synthetic and real-world data. On synthetic examples involving up to hundreds of input variables, we demonstrated its versatility and scalability under various inequality conditions. We also tested its performance on the 5D coastal flooding application in subsection 4.3.2. In contrast to experiments in Chapters 4 and 5, the additive model led to faster numerical results with a computational time limited to only dozens of seconds.

Part V

Parameter Estimation under Inequality Constraints

Chapter 7

Maximum Likelihood Estimation under Inequality Constraints

Contents

7.1 Introduction

Parts III and IV have been dedicated to the study of GP regression models under inequality constraints. There, in many of the toy examples, the covariance parameters have been either manually fixed or estimated neglecting the influence of the inequality conditions. As claimed by Maatouk et al. (2015), accounting for the constraints could lead to more reasonable estimated parameters, therefore, to models that can explain better the data. Despite the effort made in (Maatouk et al., 2015), their approach is impractical as it is based on a time-consuming cross-validation scheme. Moreover, the

optimal solution is restricted to be on a finite grid of possible values, with the same estimation of correlation parameters as for unconstrained GP parameters.

In this chapter, we aim at further investigating a maximum likelihood estimator (MLE) that accounts for inequality constraints. The main contributions are threefold. First, we suggest a constrained MLE (cMLE) under inequality conditions. Second, we show that, loosely speaking, any consistency result for ML with unconstrained GPs, is preserved when adding boundedness, monotonicity and convexity constraints. Third, we study some cases where both the MLE and cMLE are asymptotically Gaussian distributed, conditionally to the fact that the GP satisfies the constraints.

This chapter is based on the papers:

- López-Lopera, A. F., Bachoc, F., Durrande, N., and Roustant O. (2018). Finite-dimensional Gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1224–1255.

- Bachoc, F., Lagnoux, A., and López-Lopera, A. F. (2019). Maximum likelihood estimation for Gaussian processes under inequality constraints. *Electronic Journal of Statistics*, 13(2):2921–2969.

7.2 Covariance parameter estimation under inequality constraints

Here, we continue with the same notation as in subsection 2.2.3. Let $\mathcal{L}_n(\boldsymbol{\theta})$ and $\widehat{\boldsymbol{\theta}}_{\text{MLE}}$ be the (unconstrained) log-likelihood function and the MLE given by (2.7) and (2.8), respectively. Since $\mathcal{L}_n(\boldsymbol{\theta})$ itself does not take into account the inequality constraints, the estimated $\widehat{\boldsymbol{\theta}}_{\text{MLE}}$ may produce less realistic models. This claim is also supported by previous works where the need of incorporating those constraints in the covariance parameter estimation is key to properly describe data (see, e.g., Maatouk et al., 2015). Therefore, here we further investigate a constrained likelihood function that accounts for the inequality conditions.

Without loss of generality, we assume one dimensional input spaces, i.e. $x \in [0,1]$, and we consider the model of (3.1). Let $P_{\boldsymbol{\theta}}(\mathbf{Y}_m|\boldsymbol{\xi} \in \mathcal{C})$ be the conditional pdf of $\mathbf{Y}_m = \big[Y_m(x_1), \ \cdots, \ Y_m(x_n)\big]^\top$ given $\boldsymbol{\xi} \in \mathcal{C}$, when Y has covariance function $k_{\boldsymbol{\theta}}$. By using Bayes' theorem, the constrained log-likelihood $\mathcal{L}_{m,n,c}(\boldsymbol{\theta}) = \log P_{\boldsymbol{\theta}}(\mathbf{Y}_m|\boldsymbol{\xi} \in \mathcal{C})$ is

$$\mathcal{L}_{m,n,c}(\boldsymbol{\theta}) = \log P_{\boldsymbol{\theta}}(\mathbf{Y}_m) + \log P_{\boldsymbol{\theta}}(\boldsymbol{\xi} \in \mathcal{C}|\boldsymbol{\Phi}\boldsymbol{\xi} = \mathbf{Y}_m) - \log P_{\boldsymbol{\theta}}(\boldsymbol{\xi} \in \mathcal{C}), \tag{7.1}$$

where the first term is the unconstrained log-likelihood, and the last two terms depend on the inequality constraints. Then, the cMLE is given by

$$\widehat{\boldsymbol{\theta}}_{\text{cMLE}} = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}_{m,n,c}(\boldsymbol{\theta}). \tag{7.2}$$

Notice that $P_{\boldsymbol{\theta}}(\boldsymbol{\xi} \in \mathcal{C}|\boldsymbol{\Phi}\boldsymbol{\xi} = \mathbf{Y}_m)$ and $P_{\boldsymbol{\theta}}(\boldsymbol{\xi} \in \mathcal{C})$ are Gaussian orthant probabilities. As they have no explicit expressions, numerical procedures have been investigated (see,

e.g., Botev, 2017; Genz, 1992). The estimator proposed in (Genz, 1992) is based on a separation-of-variables transformation which reduces the problem to standard numerical multiple integration algorithms. On the other hand, as briefly described in subsection 3.3.2, the estimator from (Botev, 2017) efficiently deals with hitherto intractable Gaussian integrals in high dimensions via minimax exponential tilting. Due to numerical stabilities, we use (Botev, 2017) in further experiments. Hence, the likelihood evaluation and optimisation of (7.1) and (7.2) have to be done numerically.

Next, we study some asymptotic properties of the MLEs for constrained GPs, For readability and concision, we refer to Appendix A, and references (Bachoc et al., 2019; López-Lopera et al., 2018), for the proofs of the propositions, lemmas and theorems discussed in Sections 7.3 and 7.4. Furthermore, we illustrate the asymptotic properties on 1D synthetic examples.

7.3 Asymptotic consistency of maximum likelihood estimators

Now, we consider the fixed-domain asymptotic setting (Stein, 1999), with a dense sequence of observation points in a bounded domain. It should be noted that, when the GP is not constrained, significant contributions have been provided to study the consistency or asymptotic normality of the MLE (Du et al., 2009; Loh, 2005; Loh and Lam, 2000; Ying, 1993; Zhang, 2004).

For $\kappa \in \{0, 1, 2\}$, let Y be a GP with C^κ trajectories on a bounded set $\mathbb{X} \subset \mathbb{R}^d$. Let $\mathcal{E}_\kappa$ be one of the following convex sets of functions

$$\mathcal{E}_\kappa = \begin{cases} f \ : \ \mathbb{X} \to \mathbb{R}, f \text{ is } C^0 \text{ and } \forall \boldsymbol{x} \in \mathbb{X}, \ \ell \leq f(\boldsymbol{x}) \leq u & \text{if } \kappa = 0, \\ f \ : \ \mathbb{X} \to \mathbb{R}, f \text{ is } C^1 \text{ and } \forall \boldsymbol{x} \in \mathbb{X}, \ \forall i = 1, \cdots, d, \ \frac{\partial}{\partial x_i} f(\boldsymbol{x}) \geq 0 & \text{if } \kappa = 1, \\ f \ : \ \mathbb{X} \to \mathbb{R}, f \text{ is } C^2 \text{ and } \forall \boldsymbol{x} \in \mathbb{X}, \ \frac{\partial^2}{\partial \boldsymbol{x}^2} f(\boldsymbol{x}) \text{ is a non-negative definite matrix} & \text{if } \kappa = 2. \end{cases} \tag{7.3}$$

For the purpose of asymptotic analysis, we do not consider the hat basis functions anymore, and we focus on the GP Y and the observation vector

$$\mathbf{Y}_n = \big[Y(x_1), \ \cdots, \ Y(x_n)\big]^\top.$$

We study the (unconstrained) log-likelihood function based on $\log P_{\boldsymbol{\theta}}(\mathbf{Y}_n)$ and the constrained log-likelihood function based on

$$\log P_{\boldsymbol{\theta}}(\mathbf{Y}_n | Y \in \mathcal{E}_\kappa) = \log P_{\boldsymbol{\theta}}(\mathbf{Y}_n) + P_{\boldsymbol{\theta}}(Y \in \mathcal{E}_\kappa | \mathbf{Y}_n) - \log P_{\boldsymbol{\theta}}(Y \in \mathcal{E}_\kappa).$$

These quantities are more challenging to evaluate in practice than for Section 7.2, but the purpose is a theoretical analysis.

7.3.1 Asymptotic consistency of the MLE and cMLE

In Proposition 7.1, we prove that if the MLE is consistent, when considering the (unconditional) distribution of Y, then it remains consistent when conditioning to $Y \in \mathcal{E}_\kappa$.

In Proposition 7.2, we prove that, under mild conditions (such as Y being κ times continuously differentiable, Adler, 1990), implying the consistency of MLE with the (unconditional) distribution of Y, the cMLE remains consistent when adding the constraint $Y \in \mathcal{E}_\kappa$.

Condition 7.1 *Let $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{X}$. For a fixed $\kappa \in \{0, 1, 2\}$, assume one of the following conditions:*

- If $\kappa = 0$. Assume that Y has continuous trajectories. Let k be the covariance function of Y. Let

$$d_k(\boldsymbol{x}, \boldsymbol{x}') = \sqrt{k(\boldsymbol{x}, \boldsymbol{x}) + k(\boldsymbol{x}', \boldsymbol{x}') - 2k(\boldsymbol{x}, \boldsymbol{x}')}.$$

Let $N(\mathbb{X}, d_k, \ell)$ be the minimum number of balls with radius ℓ (w.r.t. the distance d_k), required to cover $\mathbb{X}$. Assume that

$$\int_0^\infty \sqrt{\log(N(\mathbb{X}, d_k, \ell))}\, d\ell < \infty. \tag{7.4}$$

Assume also that the Fourier transform $\widehat{k}$ of k satisfies

$$\exists\ P < \infty \quad \text{so that as} \quad \|w\| \to \infty, \quad \widehat{k}(w)\|w\|^P \to \infty. \tag{7.5}$$

- If $\kappa = 1$. Assume that Y has C^1 trajectories. Let $k_i^{[1]}$ be the covariance function of $\frac{\partial}{\partial x_i} Y$. Let $d_{k_i^{[1]}}$ and $N(\mathbb{X}, d_{k_i^{[1]}}, \ell)$ be defined as d_k and $N(\mathbb{X}, d_k, \ell)$ for $\kappa = 0$. Assume that

$$\int_0^\infty \sqrt{\log(N(\mathbb{X}, d_{k_i^{[1]}}, \ell))}\, d\ell < \infty, \quad \forall i = 1, \cdots, d. \tag{7.6}$$

Assume also that the Fourier transform $\widehat{k}_i^{[1]}$ of $k_i^{[1]}$ satisfies the same conditions as for $\kappa = 0$.

- If $\kappa = 2$. Assume that Y has C^2 trajectories. Let $k_{i,j}^{[2]}$ be the covariance function of $\frac{\partial^2}{\partial x_i \partial x_j} Y$. Let $d_{k_{i,j}^{[2]}}$ and $N(\mathbb{X}, d_{k_{i,j}^{[2]}}, \ell)$ be defined as d_k and $N(\mathbb{X}, d_k, \ell)$ for $\kappa = 0$. Assume that

$$\int_0^\infty \sqrt{\log(N(\mathbb{X}, d_{k_{i,j}^{[2]}}, \ell))}\, d\ell < \infty, \quad \forall i, j = 1, \cdots, d. \tag{7.7}$$

Assume also that the Fourier transform $\widehat{k}_{i,j}^{[2]}$ of $k_{i,j}^{[2]}$ satisfies the same conditions as for $\kappa = 0$.

Let us discuss Condition 7.1. For $\kappa = 0$, it is assumed that Y has continuous trajectories, which implies that the covariance function k of Y is continuous (see, e.g., Theorem 1.6 in Adler (1990) or Lemma 1 in Ibragimov and Rozanov (1978)). Hence, Condition 7.1 implies that Y is mean square continuous (Stein, 1999). Mean square continuity is perhaps a more commonly used notion than trajectory continuity in the statistical literature (Rasmussen and Williams, 2005; Santner et al., 2003a). Nevertheless, in the context of this appendix, trajectory continuity is needed to define the event $\mathcal{E}_\kappa$. We also remark that (7.4), for $\kappa = 0$, is not significantly stronger than assuming that k

is continuous. In particular this condition holds if k is α-Hölder continuous with $\alpha > 0$ (since then one can show that in this case $N(\mathbb{X}, d_k, \ell)$ is a $O(\ell^{-2d/\alpha})$ as $\ell \to 0$), which is the case for the Matérn covariance function with any smoothness parameter $\nu > 0$ (Stein, 1999). Hence, Condition 7.1 holds for the Matérn covariance function with $\nu > 0$, since then also (7.5) holds (see e.g. Stein, 1999, for the expression of the Fourier transform of the Matérn covariance function). We remark however that (7.5) does not hold for the squared exponential covariance function which Fourier transformp vanishes too fast as $w \to \infty$ (Stein, 1999).

The discussion is similar for $\kappa = 1, 2$. In these cases, Condition 7.1 implies that Y is κ times mean square differentiable and that k has partial derivatives of order 2κ. Having derivatives of order 2κ is arguably a minimal condition for mean square differentiability of order κ (Santner et al., 2003a; Stein, 1999). Furthermore, if the derivatives of order 2κ of k are Hölder continuous then (7.6) or (7.7) hold. Hence, Condition 7.1 is not significantly stronger than mean square differentiability, and holds for the Matérn covariance function with $\nu > \kappa$.

Proposition 7.1 *Let Y be a zero-mean GP on a bounded set $\mathbb{X} \subset \mathbb{R}^d$ with covariance function k satisfying Condition 7.1. Let $\boldsymbol{\Theta}$ be a compact set on $(0, \infty)^{d+1}$. Let $k_{\boldsymbol{\theta}}$ be the covariance function of $x \to \sigma Y(\ell_1 x_1, \cdots, \ell_d x_d)$ for $\boldsymbol{\theta} = (\sigma^2, \ell_1, \cdots, \ell_d) \in \boldsymbol{\Theta}$. Let $\boldsymbol{\theta}^* = (1, \cdots, 1)$. Remark that $k = k_{\boldsymbol{\theta}^*}$ and assume that $\boldsymbol{\theta}^* \in \boldsymbol{\Theta}$. Let $(\boldsymbol{x}_i)_{i \in \mathbb{N}}$ be a dense sequence in $\mathbb{X}$. Let $\boldsymbol{Y}_n = \big[Y(\boldsymbol{x}_1), \ \cdots, \ Y(\boldsymbol{x}_n)\big]^\top$. Let the log-likelihood function $\mathcal{L}_n(\boldsymbol{\theta})$ be defined as in (2.7). Let $\widehat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \ \mathcal{L}_n(\boldsymbol{\theta})$. Assume that $\forall \varepsilon > 0$,*

$$P(\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\| \geq \varepsilon) \xrightarrow[n \to \infty]{} 0.$$

Let $\kappa \in \{0, 1, 2\}$. Let $\mathcal{E}_\kappa$ be as in (7.3). Then, we have $P(Y \in \mathcal{E}_\kappa) > 0$, and thus

$$P(\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\| \geq \varepsilon \mid Y \in \mathcal{E}_\kappa) \xrightarrow[n \to \infty]{} 0.$$

Proposition 7.2 *We use the same notation and assumptions as in Proposition 7.1. Let $\kappa \in \{0, 1, 2\}$ be fixed. Let $P_{\boldsymbol{\theta}}$ be the distribution of Y with covariance function $k_{\boldsymbol{\theta}}$. Let*

$$\mathcal{L}_{n,c}(\boldsymbol{\theta}) = \mathcal{L}_n(\boldsymbol{\theta}) + \log P_{\boldsymbol{\theta}}(Y \in \mathcal{E}_\kappa | \, \boldsymbol{Y}_n) - \log P_{\boldsymbol{\theta}}(Y \in \mathcal{E}_\kappa).$$

Assume that $\forall \varepsilon > 0$ and $\forall M < \infty$,

$$P\left(\sup_{\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \geq \varepsilon} (\mathcal{L}_n(\boldsymbol{\theta}) - \mathcal{L}_n(\boldsymbol{\theta}^*)) \geq -M \right) \xrightarrow[n \to \infty]{} 0.$$

Then,

$$P\left(\sup_{\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \geq \varepsilon} (\mathcal{L}_{n,c}(\boldsymbol{\theta}) - \mathcal{L}_{n,c}(\boldsymbol{\theta}^*)) \geq -M \ \middle| \ Y \in \mathcal{E}_\kappa \right) \xrightarrow[n \to \infty]{} 0.$$

Consequently

$$\arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \ \mathcal{L}_n(\boldsymbol{\theta}) \xrightarrow[n \to \infty]{P} \boldsymbol{\theta}^*, \quad \text{and} \quad \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \ \mathcal{L}_{n,c}(\boldsymbol{\theta}) \xrightarrow[n \to \infty]{P|Y \in \mathcal{E}_\kappa} \boldsymbol{\theta}^*,$$

where $\xrightarrow[n\to\infty]{P}$ *denotes the convergence in probability under the distribution of* Y*, and* $\xrightarrow[n\to\infty]{P|Y\in\mathcal{E}_\kappa}$ *denotes the convergence in probability under the distribution of* Y *given* $Y \in \mathcal{E}_\kappa$*.*

Remark 7.1 *Propositions 7.1 and 7.2 can be extended to the case of noisy observations of GPs (or to the case of a nugget effect). More precisely, these propositions would still hold if* $\boldsymbol{Y}_n$ *was replaced by* $\boldsymbol{Y}_n+\mathbf{z}_n$*, with* $\mathbf{z}_n = (z_1, ..., z_n)^T$ *where* $z_1, ..., z_n$ *are independent, independent of* Y *and follow the* $\mathcal{N}(0, \tau^2)$ *distribution with* $\tau > 0$ *fixed, known, and does not depend on* n*. Naturally,* $\mathbf{Y}_n$ *should also be* $\mathbf{Y}_n + \mathbf{z}_n$ *in the definition of the ML and cML functions. The proofs of these adaptations of Propositions 7.1 and 7.2 would be identical to those of the original propositions. In particular, the results of (Bect et al., 2016; Kallenberg, 2002) that are used in the proofs (see Lemmas A.1 and A.2 in López-Lopera et al., 2018) still hold if* $\boldsymbol{Y}_n$ *is replaced by* $\boldsymbol{Y}_n + \mathbf{z}_n$*. Similarly, we believe that Propositions 7.1 and 7.2 can also be adapted if* τ^2 *is estimated by ML or cML.*

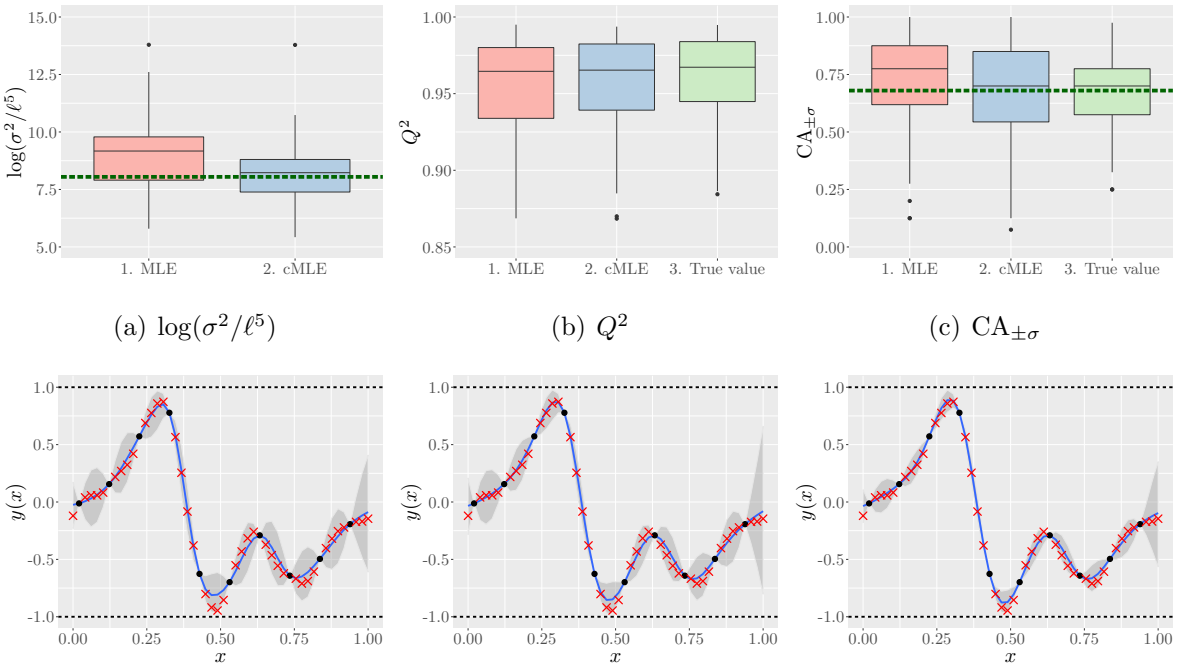
Remark 7.2 *We refer to Appendix B in (López-Lopera et al., 2018) for additional results that account for the hat basis functions in (3.1).*

7.3.2 Numerical illustration

To assess the performance of the estimator of (7.2), we simulated sample paths from a centred constrained GP Y using a Matérn 5/2 covariance function with $\boldsymbol{\theta}^* = (1, 0.2)$ (see Table 2.1). We sampled 100 realisations of Y on $\mathcal{D} = [0, 1]$ such that $Y \in [-1, 1]$. Then, for each realisation, we trained a constrained model assuming boundedness conditions with bounds $[-1, 1]$. We used 10 training points regularly spaced in $\mathcal{D}$ and $m = 50$ hat basis functions.[1] For both (unconstrained) ML and constrained ML (cML) optimisations, we used multistart with ten initial vectors of covariance parameters located on a maximin Latin hypercube DoE with $\sigma^2 \in [0, 2]$ and $\ell \in [0.04, 0.40]$. We used the `NLopt` optimisation tools from (Johnson), and we tested the different gradient-based optimisers. After some tests, we concluded that the globally-convergent method of moving asymptotes (MMA) (Svanberg, 2002) yielded more stable results for estimating the covariance parameters. For cMLE, gradients were computed numerically. As the parameters of the Matérn 5/2 kernel are non-microergodic for one-dimensional input spaces, they cannot be estimated consistently (Zhang, 2004). Therefore, we evaluated the quality of the likelihood estimators using the consistently estimable ratio $\rho = \sigma^2/\ell^5$. In Figure 7.1(a), we show the boxplots of the estimated ratios obtained with the 100 simulations drawn from the GP. Notice that the estimated logged ratios $\log \widehat{\rho}_{\text{MLE}}$ and $\log \widehat{\rho}_{\text{cMLE}}$ are reasonably close to the true value $\log \rho^* = \log(1^2/0.2^5)$, but the one using cMLE is slightly better in terms of variance and bias.

We also evaluate the efficiency of the two estimators in terms of prediction accuracy. For each realisation, we estimated the covariance parameters $\boldsymbol{\theta}^*$ by MLE and cMLE. We then simulated the posterior at 50 new regularly spaced locations using the estimated

[1] In this experiment, we manually tuned the number of basis function m. We used different values of $m = 25, 50, 100, 150, 200$, and we observed that results of Figure 7.1 remained stable after $m = 50$.



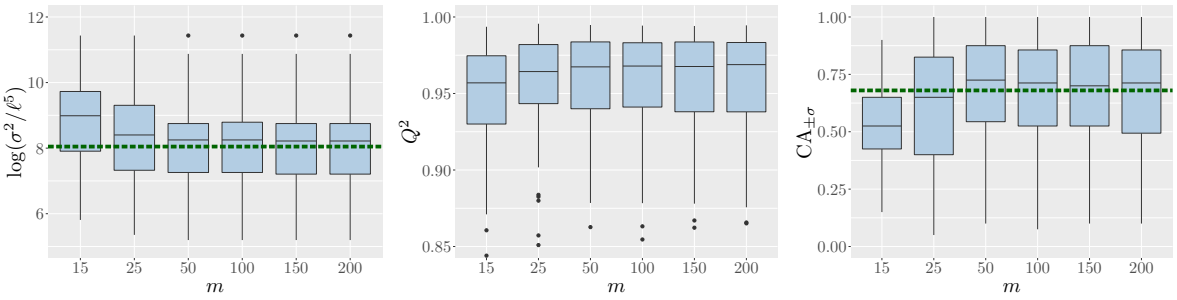
(a) $\log(\sigma^2/\ell^5)$ (b) Q^2 (c) $\mathrm{CA}_{\pm\sigma}$

(d) $\widehat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ (e) $\widehat{\boldsymbol{\theta}}_{\mathrm{cMLE}}$ (f) $\boldsymbol{\theta}^*$

Figure 7.1: Assessment of the likelihood (ML) and conditional likelihood (cML) estimators for 100 samples drawn from a GP with true parameters $\boldsymbol{\theta}^* = (1, 0.2)$, and satisfying the bounds $[-1, 1]$. (a) Estimated values of the log-ratio $\log \rho^* = \log(1^2/0.2^5)$ (dashed green line) using MLE and cMLE. Predictive accuracies are evaluated using the (b) Q^2 and (c) $\mathrm{CA}_{\pm\sigma}$ criteria. In Figure (c), the horizontal dashed green line represents the 68% pointwise coverage. Predictions are shown for one sample using (d) $\widehat{\boldsymbol{\theta}}_{\mathrm{MLE}}$, (e) $\widehat{\boldsymbol{\theta}}_{\mathrm{cMLE}}$, and (f) $\boldsymbol{\theta}^*$. For the predictions, panel description is the same as Figure 3.2.

covariance parameter $\widehat{\boldsymbol{\theta}}$. The conditional sample paths were simulated via HMC. We used the Q^2 and $\mathrm{CA}_{\pm\sigma}$ criteria to assess the quality of predictions over the 50 new values.

Figure 7.1 shows the inferred sample paths for one realisation using either $\widehat{\boldsymbol{\theta}}_{\mathrm{MLE}}$, $\widehat{\boldsymbol{\theta}}_{\mathrm{cMLE}}$, or $\boldsymbol{\theta}^*$ (Figures 7.1(d) to 7.1(f), respectively). We observe that, in the three cases, the models tend to fit properly the test data with accurate confidence intervals. According to Figures 7.1(b) and 7.1(c), we see that they provide Q^2 and $\mathrm{CA}_{\pm\sigma}$ median values close to the ones obtained when the true $\boldsymbol{\theta}^*$ is used. Although the predictive accuracies obtained using cMLE are better than for MLE in terms of bias, we observe larger variances in the $\mathrm{CA}_{\pm\sigma}$ criterion for cMLE. We also compute the lengths of the one standard deviation intervals and we observed that cMLE provides smaller intervals than the ones by MLE. This is consistent with Figure 7.1(c). Since the Gaussian orthant terms from the conditional likelihood of (7.1) have to be approximated, we believe that this affects the effectiveness of cMLE. Furthermore, existing estimators of Gaussian orthant probabilities present some numerical instabilities limiting the cML optimisation routine and providing suboptimal results. Finally, notice that MLE also provides reliable



(a) (b) (c)

Figure 7.2: Assessment of the cMLE using different number of knots m for the example in Figure 7.1. Results are shown for (a) the consistently estimable ratio $\rho = \sigma^2/\ell^5$, (b) the Q^2 criterion, and (c) $\mathrm{CA}_{\pm\sigma}$ criterion.

predictions. This suggests that, if we properly take into account the inequality constraints in the posterior distribution, the unconstrained ML optimisation can be used for practical implementation.

Finally, we test the quality of cMLE using different number of knots m. The experiment of Figure 7.1 has been repeated for $m = 15, 25, 50, 100, 150, 200$. Figure 7.2 shows the results of the consistently estimable ratio $\rho = \sigma^2/\ell^5$, the Q^2 and $\mathrm{CA}_{\pm\sigma}$ criteria. One can observe that results become stable for $m \geq 50$.

7.4 Asymptotic normality of maximum likelihood estimators

Here, we use the same notation as in Sections 7.2 and 7.3. We consider a parametric set of covariance functions $\{k_{\boldsymbol{\theta}}; \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ defined on $[0,1]^d$, where $\boldsymbol{\Theta}$ is compact in $\mathbb{R}^p$. We also assume that, for each $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, there exists a centred GP with continuous realisations having a covariance function $k_{\boldsymbol{\theta}}$. We refer to, e.g., (Adler, 1990) for mild smoothness conditions on $k_{\boldsymbol{\theta}}$ ensuring this. We assume that the information $Y \in \mathcal{E}_\kappa$ is available where $\mathcal{E}_\kappa$ is a convex set of functions defined in (7.3).

We consider a triangular array $(\boldsymbol{x}_i^{(n)})_{n\in\mathbb{N}, i=1,\ldots,n}$ of observation points in $[0,1]^d$, where we write for concision $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = (\boldsymbol{x}_1^{(n)}, \ldots, \boldsymbol{x}_n^{(n)})$. We assume that $(\boldsymbol{x}_i^{(n)})$ is dense, that is $\sup_{\boldsymbol{x}\in[0,1]^d} \inf_{i=1,\ldots,n} |\boldsymbol{x} - \boldsymbol{x}_i^{(n)}| \to 0$ as $n \to \infty$. In some cases, we need to assume that as $n \to \infty$, the triangular array contains finer and finer tensorised grids. In some other cases, we only need to assume that the observation points constitute a sequence.

Condition 7.2 *There exist d sequences $(v_i^{(j)})_{i\in\mathbb{N}}$ for $j = 1, \ldots, d$ that are dense in $[0,1]$ and so that for all $N \in \mathbb{N}$, there exists $n_0 \in \mathbb{N}$ such that for $n \geq n_0$ we have $\{(v_{i_1}^{(1)}, \ldots, v_{i_d}^{(d)}),\ 1 \leq i_1, \ldots, i_d \leq N\} \subset (x_i)_{i=1,\ldots,n}$.*

In our opinion, Condition 7.2 is reasonable and natural. Its purpose is to guarantee that the partial derivatives of Y are consistently estimable from $\mathbf{Y}_n$ everywhere on $[0,1]^d$ (see, for instance, the proof of Theorems 7.1 and 7.2 for $\kappa = 1$ in Bachoc et al., 2019). We believe that, for the results for which Condition 7.2 is assumed, one could replace it by a milder condition and prove similar results. Then the proofs would be based on essentially the same ideas as the current ones, but could be more cumbersome.

Condition 7.3 *For all $n \in \mathbb{N}$ and $i \leq n$, we have $x_i^{(n)} = x_i^{(i)}$.*

Condition 7.3 implies that sequences of conditional expectations w.r.t. the observations are martingales. This is necessary in some of the proofs (for instance, that of Theorem 7.2) where convergence results for martingales are used.

Next, we study the asymptotic conditional distributions of MLE and cMLE, given that the GP satisfies the constraint $Y \in \mathcal{E}_\kappa$. We consider the estimation of a single variance parameter and the estimation of the microergodic parameter in the isotropic Matérn family of covariance functions. In both cases, we show that the asymptotic conditional distributions of both estimators are identical to the unconditional asymptotic distribution of MLE. Hence, it turns out that the impact of the constraints on covariance parameter estimation is asymptotically negligible. In simulations, we confirm that for large sample sizes, the estimators have very similar empirical distributions, that are close to the asymptotic Gaussian distribution.

7.4.1 Variance parameter estimation

Here, we focus on the estimation of a single variance parameter when the correlation function is known. Hence, we let $p = 1$, $\boldsymbol{\theta} = \sigma^2$, and for $\boldsymbol{x}, \boldsymbol{x}' \in [0,1]^d$,

$$k_{\sigma^2}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 k_1(\boldsymbol{x}, \boldsymbol{x}'), \tag{7.8}$$

where k_1 is a fixed known correlation function. We make the following assumption.

Condition 7.4 *Let κ be fixed in $\{0,1,2\}$.*

- *k_1 is stationary, that is $k_1(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x} - \boldsymbol{x}')$ for $\boldsymbol{x}$ and $\boldsymbol{x}'$ in $[0,1]^d$.*
- *If $\kappa = 0$, k_1 is α-Hölder, which means that there exist nonnegative constants C and α such that*

  $$|k_1(\boldsymbol{t}) - k_1(\boldsymbol{t}')| \leq C||\boldsymbol{t} - \boldsymbol{t}'||^\alpha$$

  for all $\boldsymbol{t}$ and $\boldsymbol{t}'$ in $[0,1]^d$, where $||\cdot||$ is the Euclidean norm. Furthermore, the Fourier transform $\widehat{k}_1$ of k_1 satisfies, for some fixed $P < \infty$,

  $$\widehat{k}_1(\omega)||\omega||^P \underset{\omega \to \infty}{\longrightarrow} \infty. \tag{7.9}$$

- *If $\kappa = 1$, the GP Y is differentiable in quadratic mean. For $i = 1, \ldots, d$, let $k_{1,i}$ be the covariance function of $\partial Y / \partial x_i$. Then $k_{1,i}$ is α-Hölder for a fixed $\alpha > 0$. Also, (7.9) holds with $\widehat{k}_1$ replaced by the Fourier transform $\widehat{k}_{1,i}$ of $k_{1,i}$ for $i = 1, \ldots, d$.*

- *If $\kappa = 2$, the GP Y is twice differentiable in quadratic mean. For $i, j = 1, \ldots, d$, let $k_{2,i,j}$ be the covariance function of $\partial^2 Y/(\partial x_i \partial x_j)$. Then $k_{2,i,j}$ is α-Hölder for a fixed $\alpha > 0$. Also, (7.9) holds with $\widehat{k}_1$ replaced by the Fourier transform $\widehat{k}_{1,i,j}$ of $k_{1,i,j}$ for $i, j = 1, \ldots, d$.*

These conditions make the conditioning by $Y \in \mathcal{E}_\kappa$ valid for $\kappa = 0, 1, 2$ as established in the following lemma.

Lemma 7.1 *Assume that Condition 7.4 holds. Then for all $\kappa \in \{0, 1, 2\}$ and for any compact K in $(0, +\infty)$, we have*

$$\inf_{\sigma^2 \in K} P_{\sigma^2}(Y \in \mathcal{E}_\kappa) > 0.$$

Now, we first study the conditional asymptotic distribution of the (unconstrained) MLE conditionally to $Y \in \mathcal{E}_\kappa$. Then we study the cMLE of Proposition 7.2

Asymptotic conditional distribution of the maximum likelihood estimator: the log-likelihood function in Proposition 7.1 for σ^2 can be written as

$$\mathcal{L}_n(\sigma^2) = -\frac{n}{2}\log(\sigma^2) - \frac{1}{2}\log(\det(\boldsymbol{R}_1)) - \frac{1}{2\sigma^2}\mathbf{Y}_n^\top \boldsymbol{R}_1^{-1}\mathbf{Y}_n - \frac{n}{2}\log(2\pi), \tag{7.10}$$

where $\boldsymbol{R}_1 = (k_1(x_i, x_j))_{1 \le i,j \le n}$. Then the standard MLE is given by

$$\widehat{\sigma}_n^2 \in \arg\max_{\sigma^2 > 0} \mathcal{L}_n(\sigma^2). \tag{7.11}$$

According to (Bachoc et al., 2019), one can show that, for $\kappa = 0, 1, 2$, $\sqrt{n}(\widehat{\sigma}_n^2 - \sigma_0^2)$ is asymptotically Gaussian distributed conditionally to $Y \in \mathcal{E}_\kappa$.

Theorem 7.1 *For $\kappa = 1, 2$, we assume that Condition 7.2 holds. Under Condition 7.4, the MLE $\widehat{\sigma}_n^2$ of σ_0^2 defined by (7.11) conditioned on $Y \in \mathcal{E}_\kappa$ is asymptotically Gaussian distributed. More precisely, for $\kappa = 0, 1, 2$,*

$$\sqrt{n}\left(\widehat{\sigma}_n^2 - \sigma_0^2\right) \xrightarrow[n \to +\infty]{\mathcal{L}|Y \in \mathcal{E}_\kappa} \mathcal{N}(0, 2\sigma_0^4).$$

It is well known that $\sqrt{n}(\widehat{\sigma}_n^2 - \sigma_0^2)$ converges (unconditionally) to the $\mathcal{N}(0, 2\sigma_0^4)$ distribution (Stein, 1999). Hence, conditioning by $Y \in \mathcal{E}_\kappa$ has no impact on the asymptotic distribution of the MLE.

Asymptotic conditional distribution of the constrained maximum likelihood estimator: we assume that the compact set $\boldsymbol{\Theta}$ is $[\sigma_l^2, \sigma_u^2]$ with $0 < \sigma_l^2 < \sigma_0^2 < \sigma_u^2 < +\infty$, and we consider the cMLE $\widehat{\sigma}_{n,c}^2$ of σ_0^2 derived by maximizing on the compact set $\boldsymbol{\Theta}$ the constrained log-likelihood in Proposition 7.2:

$$\widehat{\sigma}_{n,c}^2 \in \arg\max_{\sigma^2 \in \Theta} \mathcal{L}_{n,c}(\sigma^2). \tag{7.12}$$

Then, one can show that the conditional asymptotic distribution of the cMLE is the same as that of the MLE (see, e.g., Bachoc et al., 2019, for further discussion).

Theorem 7.2 *For $\kappa = 1, 2$, we assume that Condition 7.2 holds. Under Condition 7.3 and 7.4, the cMLE $\widehat{\sigma}^2_{n,c}$ of σ_0^2 defined in (7.12) is asymptotically Gaussian distributed. More precisely, for $\kappa = 0, 1, 2$,*

$$\sqrt{n}\left(\widehat{\sigma}^2_{n,c} - \sigma_0^2\right) \xrightarrow[n\to+\infty]{\mathcal{L}|Y\in\mathcal{E}_\kappa} \mathcal{N}(0, 2\sigma_0^4).$$

7.4.2 Microergodic parameter estimation for the isotropic Matérn model

In this section, we let $d = 1, 2$ or 3, and we consider the isotropic Matérn family of covariance functions on $[0,1]^d$. We refer to, e.g., Stein (1999) for more details. Here $k_{\boldsymbol{\theta}} = k_{\boldsymbol{\theta},\nu}$ is given by, for $\boldsymbol{x}, \boldsymbol{x}' \in [0,1]^d$,

$$k_{\boldsymbol{\theta},\nu}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 K_\nu\left(\frac{||\boldsymbol{x} - \boldsymbol{x}'||}{\ell}\right) = \frac{\sigma^2}{\Gamma(\nu)2^{\nu-1}}\left(\frac{||\boldsymbol{x} - \boldsymbol{x}'||}{\ell}\right)^\nu \kappa_\nu\left(\frac{||\boldsymbol{x} - \boldsymbol{x}'||}{\ell}\right).$$

The parameter $\sigma^2 > 0$ is the variance of the process, $\ell > 0$ is the length-scale parameter that controls how fast the covariance function decays with the distance, and $\nu > 0$ is the regularity parameter of the process. The function κ_ν is the modified Bessel function of the second kind of order ν (see Abramowitz and Stegun, 1964). We assume in the sequel that the smoothness parameter ν is known. Then $\boldsymbol{\theta} = (\sigma^2, \ell)$ and $p = 2$.

Condition 7.5 *For $\kappa = 0$ (respectively $\kappa = 1$ and $\kappa = 2$), we assume that $\nu > 0$ (resp. $\nu > 1$ and $\nu > 2$).*

We remark that Condition 7.5 naturally implies Condition 7.4 so that the conditioning by $Y \in \mathcal{E}_\kappa$ is valid for any $\kappa = 0, 1, 2$ as established in the next lemma.

Lemma 7.2 *Assume that Condition 7.5 holds. Then for all $\kappa \in \{0, 1, 2\}$ and for any compact K of $(0, \infty)^2$, we have*

$$\inf_{(\sigma^2,\ell)\in K} P_{\sigma^2,\ell}\left(Y \in \mathcal{E}_\kappa\right) > 0.$$

We refer to (Stein, 1999) for a reference on the impact of ν on the smoothness of the Matérn covariance function and on its Fourier transform.

Asymptotic conditional distribution of the maximum likelihood estimator: the log-likelihood function in Proposition 7.1 for σ^2 and ℓ under the Matérn model with fixed parameter ν can be written as

$$\mathcal{L}_n(\sigma^2, \ell) = -\frac{1}{2}n\log(\sigma^2) - \frac{1}{2}\log(\det(\boldsymbol{R}_{\ell,\nu})) - \frac{1}{2\sigma^2}\mathbf{Y}_n^\top \boldsymbol{R}_{\ell,\nu}^{-1}\mathbf{Y}_n - \frac{n}{2}\log 2\pi, \tag{7.13}$$

where $\boldsymbol{R}_{\ell,\nu} = (K_\nu(||\boldsymbol{x}_i - \boldsymbol{x}_j||/\ell))_{1\le i,j\le n}$. Let $\boldsymbol{\Theta} = [\sigma_l^2, \sigma_u^2] \times [\ell_l, \ell_u]$ with fixed $0 < \sigma_l^2 < \sigma_u^2 < \infty$ and fixed $0 < \ell_l < \ell_u < \infty$. Assume moreover that the true parameters are such that $\sigma_l^2/(\ell_l^{2\nu}) < \sigma_0^2/(\ell_0^{2\nu}) < \sigma_u^2/(\ell_u^{2\nu})$. Then, the MLE is given by

$$(\widehat{\sigma}_n^2, \widehat{\ell}_n) \in \arg\max_{(\sigma^2,\ell)\in\boldsymbol{\Theta}} \mathcal{L}_n(\sigma^2, \ell). \tag{7.14}$$

It has been shown in (Zhang, 2004) that the parameters σ_0^2 and ℓ_0 cannot be estimated consistently but that the microergodic parameter $\sigma_0^2/\ell_0^{2\nu}$ can. Furthermore, it is shown in (Kaufman and Shaby, 2013) that $\sqrt{n}(\widehat{\sigma}_n^2/\widehat{\ell}_n^{2\nu} - \sigma_0^2/\ell_0^{2\nu})$ converges to a $\mathcal{N}(0, 2\,(\sigma_0^2/\ell_0^{2\nu})^2)$ distribution. In the next theorem, we show that this asymptotic normality also holds conditionally to $Y \in \mathcal{E}_\kappa$.

Theorem 7.3 *For $\kappa = 1, 2$, we assume that Condition 7.2 holds. Under Condition 7.5, the estimator $\widehat{\sigma}_n^2/\widehat{\ell}_n^{2\nu}$ of the microergodic parameter $\sigma_0^2/\ell_0^{2\nu}$ defined by (7.14) and conditioned on $Y \in \mathcal{E}_\kappa$ is asymptotically Gaussian distributed. More precisely, for $\kappa = 0, 1, 2$,*

$$\sqrt{n}\left(\frac{\widehat{\sigma}_n^2}{\widehat{\ell}_n^{2\nu}} - \frac{\sigma_0^2}{\ell_0^{2\nu}}\right) \xrightarrow[n\to+\infty]{\mathcal{L}|Y\in\mathcal{E}_\kappa} \mathcal{N}\left(0, 2\left(\frac{\sigma_0^2}{\ell_0^{2\nu}}\right)^2\right).$$

Asymptotic conditional distribution of the constrained maximum likelihood estimator: we turn to the constrained log-likelihood and its maximizer. We consider two types of estimation settings obtained by maximizing the constrained log-likelihood $\mathcal{L}_{n,c}(\boldsymbol{\theta})$ in Proposition 7.2 under the Matérn model. In the first setting, $\ell = \ell_1$ is fixed and Proposition 7.2 is maximised over σ^2 (in the case $\ell_1 = \ell_0$ this setting is already covered by Theorem 7.2). In the second setting, Proposition 7.2 is maximised over both σ^2 and ℓ. Under the two settings, we show that the cMLE has the same asymptotic distribution as the MLE, conditionally to $Y \in \mathcal{E}_\kappa$.

Theorem 7.4 (Fixed length-scale parameter) *For $\kappa = 1, 2$, we assume that Condition 7.2 holds. Assume that Condition 7.3 and 7.5 hold. Let for $\ell \in [\ell_l, \ell_u]$,*

$$\widehat{\sigma}_{n,c}^2(\ell) \in \arg\max_{\sigma^2 \in [\sigma_l^2, \sigma_u^2]} \mathcal{L}_{n,c}(\sigma^2, \ell). \tag{7.15}$$

Let $\ell_1 \in [\ell_l, \ell_u]$ be fixed. Then $\widehat{\sigma}_{n,c}^2(\ell_1)$ is asymptotically Gaussian distributed. More precisely, for $\kappa = 0, 1, 2$,

$$\sqrt{n}\left(\frac{\widehat{\sigma}_{n,c}^2(\ell_1)}{\ell_1^{2\nu}} - \frac{\sigma_0^2}{\ell_0^{2\nu}}\right) \xrightarrow[n\to+\infty]{\mathcal{L}|Y\in\mathcal{E}_\kappa} \mathcal{N}\left(0, 2\left(\frac{\sigma_0^2}{\ell_0^{2\nu}}\right)^2\right).$$

Theorem 7.5 (Estimated length-scale parameter) *Assume that Condition 7.5 holds. Let $\widehat{\sigma}_{n,c}^2(\ell)$ be defined as in (7.15) and let $(\widehat{\sigma}_{n,c}^2, \widehat{\ell}_{n,c})$ be defined by*

$$(\widehat{\sigma}_{n,c}^2, \widehat{\ell}_{n,c}) \in \arg\max_{(\sigma^2,\ell)\in\Theta} \mathcal{L}_{n,c}(\sigma^2, \ell).$$

Notice that $\widehat{\sigma}_{n,c}^2 = \widehat{\sigma}_{n,c}^2(\widehat{\ell}_{n,c})$.

(i) For $\kappa = 0$, assume that one of the following two conditions hold.

a) We have $\nu > 1$, $d = 1$ and $\max_{x\in[0,1]} \min_{i=1,\dots,n} |x - x_i| = o(1/\sqrt{n})$.

b) We have $\nu > 2$ *and there exists a sequence* $(a_n)_{n\in\mathbb{N}}$ *with* $a_n = o(1/n^{1/4})$ *as* $n \to \infty$*, so that, for all* $\boldsymbol{x} \in [0,1]^d$*, there exists* $d+1$ *points* $v_1, \ldots, v_{d+1}$ *with* $\{v_1, \ldots, v_{d+1}\} \subset \{x_1, \ldots, x_n\}$*, so that* $\boldsymbol{x}$ *belongs to the convex hull of* $v_1, \ldots, v_{d+1}$ *and* $\max_{j=1,\ldots,d+1} ||\boldsymbol{x} - \boldsymbol{v}_j|| \leq a_n$.

(ii) For $\kappa = 1, 2$*, assume that one of the following two conditions hold.*

a) We have $\nu > \kappa + 1$*,* $d = 1$ *and* $\max_{x\in[0,1]} \min_{i=1,\ldots,n} |x - x_i| = o(1/\sqrt{n})$.

b) We have $\nu > \kappa + 2$ *and the observation points* $\{x_1, \ldots, x_n\}$ *are so that, for all* $n \geq 2^d$*, with* $N = \lfloor n^{1/d} \rfloor$*,*

$$\{x_1, \ldots, x_n\} \supset \left\{ \left(\frac{i_1}{N-1}, \ldots, \frac{i_d}{N-1} \right),\ 0 \leq i_1, \ldots, i_d \leq N-1 \right\}.$$

Then, $\widehat{\sigma}^2_{n,c}/\widehat{\ell}^{2\nu}_{n,c}$ *is asymptotically Gaussian distributed. More precisely, for* $\kappa = 0, 1, 2$*,*

$$\sqrt{n}\left(\frac{\widehat{\sigma}^2_{n,c}}{\widehat{\ell}^{2\nu}_{n,c}} - \frac{\sigma_0^2}{\ell_0^{2\nu}} \right) \xrightarrow[n\to+\infty]{\mathcal{L}|Y\in\mathcal{E}_\kappa} \mathcal{N}\left(0, 2\left(\frac{\sigma_0^2}{\ell_0^{2\nu}}\right)^2\right).$$

In Theorem 7.5, we assume that ν is larger than in Condition 7.5, and we assume that the observation points have specific quantitative space filling properties. The condition i) b) also implies that a portion of the observation points are located in the corners and borders of $[0,1]^d$. Furthermore, the condition ii) b) implies that the majority of the observation points are located on regular grids. We believe that these two last conditions could be replaced by milder ones, at the cost of proofs similar to but more cumbersome than the present ones.

We make stronger assumptions in Theorem 7.5 than in Theorem 7.4 because the former is more challenging than the latter. Indeed, since $\ell = \ell_1$ is fixed in Theorem 7.4, we can use the equivalence of two fixed Gaussian measures in order to obtain asymptotic properties of the conditional mean function of Y under $k_{1,\ell_1,\nu}$ (see the developments in the proofs in Bachoc et al., 2019). This is not possible anymore when considering the conditional mean function of Y under $k_{1,\widehat{\ell}_{n,c},\nu}$, where $\widehat{\ell}_{n,c}$ is random. Hence, we use other proof techniques, based on reproducing kernel Hilbert spaces, for studying this conditional mean function, for which the above additional conditions are needed. We refer for instance to the developments in the supplementary material in (Bachoc et al., 2019) for more details.

7.4.3 Numerical illustration

Now, we illustrate numerically the conditional asymptotic normality of the MLE and cMLE of the microergodic parameter for the Matérn 5/2 covariance function. We let $d = 1$, $m = 300$, and $x_1, ..., x_n$ be equispaced in $[0,1]$ in the rest of subsection. Since the event $Y \in \mathcal{E}_\kappa$ cannot be simulated exactly in practice, we consider the piecewise affine interpolation Y_m of Y at $t_1, \ldots, t_m \in [0,1]$, with $m > n$ (see Sections 3.2 and 7.2). Then,

the event $Y \in \mathcal{E}_\kappa$ is approximated by the event $Y_m \in \mathcal{E}'_\kappa$, where $\mathcal{E}'_0$ (respectively $\mathcal{E}'_1$, $\mathcal{E}'_2$) is the set of continuous bounded between l and u (resp. increasing, convex) functions. We can efficiently sample Y_m conditionally to $Y_m \in \mathcal{E}'_\kappa$ via HMC (Pakman and Paninski, 2014).

We consider the Matérn 5/2 covariance function $k_{\boldsymbol{\theta},5/2}$ with $\boldsymbol{\theta} = (\sigma^2, \ell)$. Here, use the parametrisation of Table 2.1 rather than that of subsection 7.4.2. For an easy reading, we keep the same notation.

Numerical results when ℓ_0 is known: for $\kappa = 0, 1$, we generate $N = 1,000$ trajectories of Y_m given $Y_m \in \mathcal{E}'_\kappa$. For each trajectory, we compute the estimators of the variance parameter $\widehat{\sigma}^2_{m,n}$ and $\widehat{\sigma}^2_{m,n,c}$ resulting by ML and cML, respectively.

In Figure 7.3, we report the results for $\kappa = 0$ (boundedness constraints) with $(\sigma_0^2, \ell_0) = (2, 0.2)$ and $n = 20, 50, 80$. We show the probability density functions obtained from the samples $\left\{n^{1/2}(\widehat{\sigma}^2_{m,n}(\ell_0)_i - \sigma_0^2)\right\}_{i=1,\cdots,N}$ and $\left\{n^{1/2}(\widehat{\sigma}^2_{m,n,c}(\ell_0)_i - \sigma_0^2)\right\}_{i=1,\cdots,N}$ obtained as discussed above. We also plot the probability density function of the limit $\mathcal{N}(0, 2\sigma_0^4)$ distribution. We observe that for a small number of observations, e.g. $n = 20$, the distribution of the cMLE is closer to the limit distribution than that of the MLE in terms of median value. We also observe that, as n increases, both distributions become more similar to the limit one. Nevertheless, the cMLE exhibits faster convergence.

In Figure 7.4, we report the same quantities for $\kappa = 1$ (monotonicity constraints) and for $(\sigma_0^2, \ell_0) = (0.5^2, 1)$. In this case, we observe that the distributions of both the MLE and the cMLE are close to the limit one even for small values of n ($n = 5, 20$).

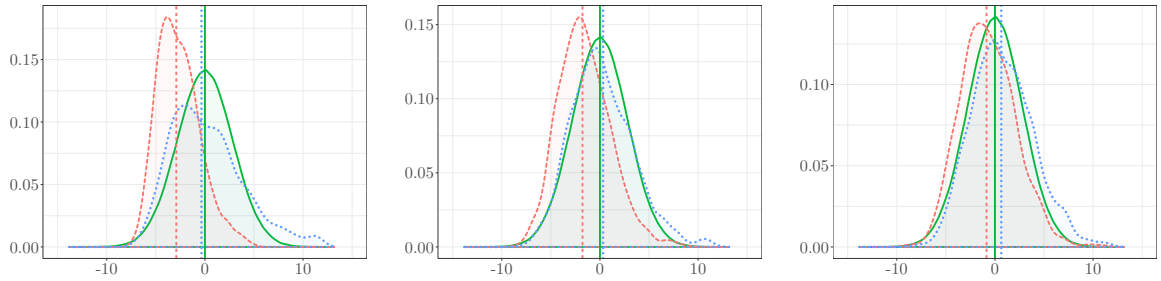
Numerical results when ℓ_0 is unknown: we let $\kappa = 0$, $(\sigma_0^2, \ell_0) = (2, 0.2)$ and $n = 20, 50, 80$. We proceed similarly as in the case where ℓ_0 is known, and we estimate the covariance parameters $(\widehat{\sigma}^2_{m,n}, \widehat{\ell}_{m,n})$ and $(\widehat{\sigma}^2_{m,n,c}, \widehat{\ell}_{m,n,c})$ via ML and cML, respectively.

In Figure 7.5, we show the probability density functions obtained from the samples $\{n^{1/2}(\widehat{\sigma}^2_{m,n,i}/\widehat{\ell}^5_{m,n,i} - \sigma_0^2/\ell_0^5)\}_{i=1,\ldots,N}$ and $\{n^{1/2}(\widehat{\sigma}^2_{m,n,c,i}/\widehat{\ell}^5_{m,n,c,i} - \sigma_0^2/\ell_0^5)\}_{i=1,\ldots,N}$, with $N = 1,000$. One can observe that the distribution of the cMLE tends to be closer to the limit one, than that of the MLE. Moreover, the convergence with the cMLE is faster than with the MLE in terms of median value.

7.5 2D application: nuclear safety criticality

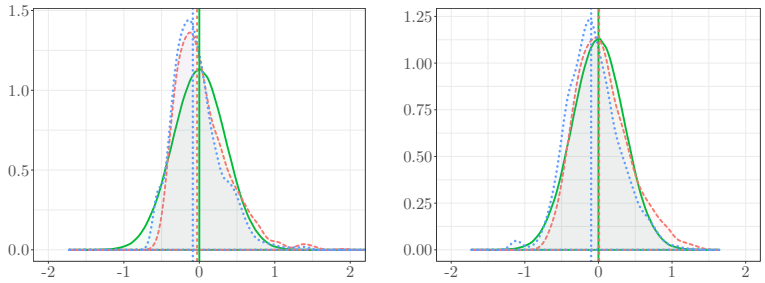
Here, we revisit the 2D nuclear example of Section 3.4. We repeat the experiment proposed there for the constrained GP model accounting both positive and non-decreasing conditions, but estimating the covariance parameters via cML. We test both unconstrained and constrained models with twenty different random Latin hypercube DoEs using several values of n. We used the Q^2 and $\mathrm{CA}_{\pm\sigma}$ criteria to evaluate the quality of the predictions.

Figure 7.6 shows that the constrained models often outperform the unconstrained ones. Notice that although the Q^2 results obtained by the unconstrained model are



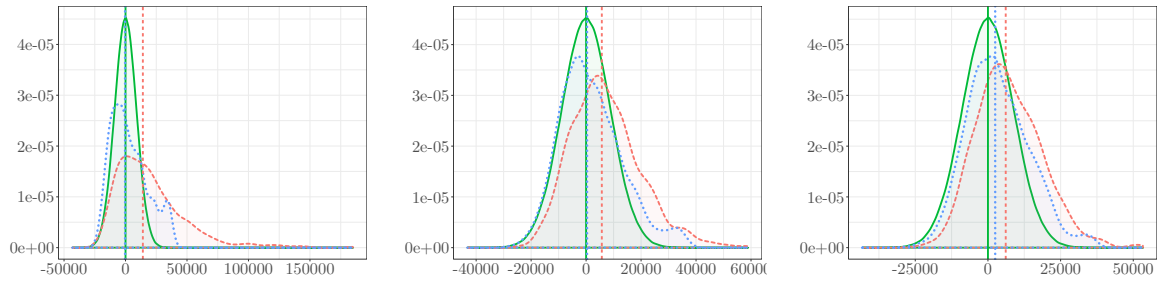
(a) $n = 20$ (b) $n = 50$ (c) $n = 80$

Figure 7.3: Asymptotic conditional distribution of the variance parameter estimators under boundedness constraints. Here $(\sigma_0^2, \ell_0) = (2, 0.2)$. Each panel shows: the limit conditional distribution $\mathcal{N}(0, 2\sigma_0^4)$ (solid), the distributions of the MLE (dashed) and of the cMLE (dotted). Vertical lines represent the median values of the distributions. Each subcaption shows the number of observations n used for the estimations.



(a) $n = 5$ (b) $n = 20$

Figure 7.4: Asymptotic conditional distribution of the variance parameter estimators under monotonicity constraints. Here $(\sigma_0^2, \ell_0) = (0.5^2, 1)$. Panel description is the same as in Figure 7.3.



(a) $n = 20$ (b) $n = 50$ (c) $n = 80$

Figure 7.5: Asymptotic conditional distribution of the microergodic parameter estimators for the isotropic Matérn 5/2 model under boundedness constraints. Panel description is the same as in Figure 7.3, with $\mathcal{N}(0, 2\sigma_0^4)$ replaced by $\mathcal{N}(0, 2(\sigma_0^2/\ell_0^5)^2)$.

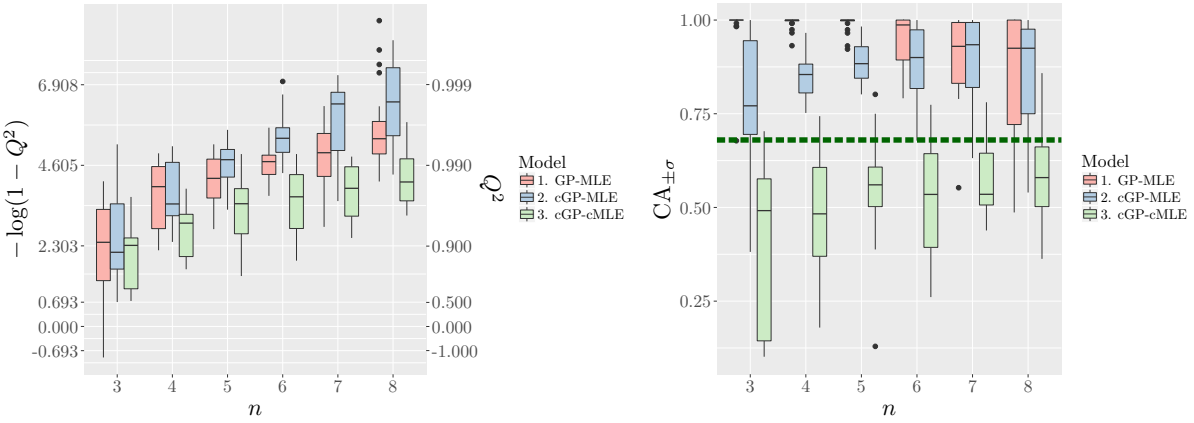


Figure 7.6: Assessment of the Gaussian models for interpolating the dataset from Figure 3.6 using various number of training points n and using twenty different random Latin hypercube designs. Predictive accuracy is evaluated using the (left) Q^2 and (right) $\mathrm{CA}_{\pm\sigma}$ criteria. Results are shown for the unconstrained GP using MLE, and the constrained GPs (cGPs) using either MLE or cMLE.

comparable with the constrained ones when the number of training points is large enough, we observe, according to the $\mathrm{CA}_{\pm\sigma}$ criterion, that the constrained model using cMLE provides more reliable confidence intervals. This means that, if we consider both positivity and monotonicity conditions to take into account the physics of the k_{eff} factor, we can obtain more informative and robust models. Furthermore, we have to note that the unconstrained MLE achieves a good tradeoff between prediction accuracy/reliability and computational cost.

7.6 Conclusions

We have further investigated a constrained likelihood that account the inequality conditions. On a 1D simulation example with known covariance parameters, we observed that the cMLE provided more consistent estimations of the consistently estimable ratio of Matérn 5/2 kernels compared to the unconstrained one. Moreover, constrained models together with the cMLE leaded to improvements of the prediction accuracy in both the predictive mean and predictive variances. In particular, the full constrained framework provided thinner confidence intervals while ensuring reliable coverage accuracies of data. The latter conclusion was also observed on the 2D nuclear application of Section 3.4. Finally, we noted that MLE can also be used yielding a good tradeoff between prediction accuracy/reliability and computational cost.

We showed that, loosely speaking, any consistency result for ML with unconstrained GPs, is preserved when adding boundedness, monotonicity and convexity constraints. Furthermore, this consistency occurs for both the unconditional and conditional likelihood functions. We remark that, under the fixed domain asymptotic framework we

consider, some covariance parameters (for instance σ^2 and ℓ in subsection 7.3.2) cannot be estimated consistently and do not have an asymptotic impact on prediction and conditional distributions. These results appear in the literature for unconstrained Gaussian processes (Stein, 1999; Zhang, 2004). We believe that it can be shown that, roughly speaking, these results also hold for constrained GPs. Hence, from an asymptotic point of view, covariance parameters that cannot be estimated consistently have an asymptotically negligible impact on prediction, also for constrained GPs.

We also showed that both the MLE and cMLE are asymptotically Gaussian distributed, conditionally to the fact that the GP satisfies either boundedness, monotonicity or convexity constraints. Their asymptotic distributions are identical to the unconditional asymptotic distribution of the MLE. In simulations, we demonstrated that the estimators have very similar performances when the number n of observations becomes large enough. We also observed that the cMLE is more accurate for small or moderate values of n.

As discussed throughout this chapter, the constrained estimator suffers from certain drawbacks that limit its practical implementations. First, since there is no a closed form of the constrained likelihood, the evaluation and optimisation of cMLE have to be done numerically. Second, it requires the computation of Gaussian orthant probabilities in high dimensions. To the best of our knowledge, existing methods for approximating those probabilities are time-consuming and present unstable results. Hence, a further investigation is needed for addressing these limitations.

Part VI

Spatial Processes under Inequality Constraints

Chapter 8

Cox Processes under Inequality Constraints

Contents

8.1 Introduction

In previous chapters, we studied a full GP-based model where inequality constraints are taken into account for data interpolation, uncertainty quantification and covariance parameter estimation. On both synthetic and real-world examples, we showed that the proposed framework can be successfully used as a stochastic emulator leading to more realistic models guided by the physics of data. Moreover, it can be coupled to other types of GP-modulated stochastic processes where constrained Gaussian priors are necessary

to get accurate models with more realistic uncertainties.

In this chapter, we adapt our approach to modelling Poisson-based point processes under inequality constraints. More precisely, we introduce a novel finite-dimensional approximation of GP-modulated Cox processes where positiveness conditions can be imposed directly on the GP, with no restrictions on the covariance function. The proposed approach can also ensure other types of inequality constraints (e.g. monotonicity, convexity), resulting in more versatile models that can be used for other classes of point processes (e.g. renewal processes). We demonstrate on both synthetic and real-world data that the proposed framework leads to accurate inference results that are competitive with those provided by other methods from the state-of-the-art.

This chapter is based on a joint work with PROWLER.io:

- López-Lopera, A. F., John, S., and Durrande, N. (2019). Gaussian process modulated Cox processes under linear inequality constraints. In *AISTATS*, 22:1997–2006.

8.2 Point Poisson processes

Poisson processes are the foundation for modelling point patterns (Kingman, 1992). They have been used in a great variety of real-world problems for modelling temporal and spatiotemporal point patterns in diverse fields such as astronomy, biology, and ecology (Baddeley et al., 2015; Møller and Waagepetersen, 2004). In reliability analysis, they are used as renewal processes to model the lifetime of items or failure (hazard) rates (Cha and Finkelstein, 2018).

Let us first summarise some concepts of spatial point processes before introducing formally the definition of a (spatial) point Poisson process. Consider a random number of events $N_B := N(B)$ occurring in a subset $B \subseteq \mathcal{S}$ with $\mathcal{S} \subseteq \mathbb{R}^d$. Then N_B is Poisson-distributed if for all $n \in \mathbb{N}$,

$$P(N_B = n) = \exp(-\mu_B)\frac{\mu_B^n}{n!}, \tag{8.1}$$

with average number of events μ_B. Then, we can write

$$N_B \sim \text{Pois}(\mu_B).$$

One property of Poisson-distributed random variables is that both the expected value and variance are equal to the rate μ_B, i.e. $\mathbb{E}\{N_B\} = \text{var}\{N_B\} = \mu_B$. Here, the constant μ_B is the average number of events, however it can also be an intensity measure parameter.

Definition 8.1 (Intensity measure) *Consider the space $\mathcal{S} \subseteq \mathbb{R}^d$ and an intensity function $\lambda : \mathcal{S} \to [0, \infty]$ that is locally integrable $\int_B \lambda(\boldsymbol{x})d\boldsymbol{x} < \infty$ for all bounded $B \subseteq \mathcal{S}$. One defines an intensity measure μ_B by*

$$\mu_B = \int_B \lambda(\boldsymbol{x})d\boldsymbol{x}, \tag{8.2}$$

and furthermore, one assume that this measure is locally finite, i.e. $\mu_B < \infty$ for $B \subseteq \mathcal{S}$.

A spatial Poisson process is a special case of point processes where there is no interaction between the point patterns (or events), i.e. "complete spatial randomness".

Definition 8.2 (Binomial point process) *Let f be a probability density function on a set $B \subseteq \mathcal{S}$ and let $n \in \mathbb{N}$. A point process X consisting of n iid points with common density f is called a binomial point process of n points in B with density f:*

$$X \sim \text{binomial}\,(B, n, f)\,.$$

Definition 8.3 (Spatial Poisson process) *A point process X on $\mathcal{S} \subseteq \mathbb{R}^d$ is a Poisson (point) process with intensity function λ, and intensity measure μ as in Definition 8.1, if the two following properties hold:*

1. *for any $B \subseteq \mathcal{S}$ with $\mu_B < \infty$, the number of events N_B is Poisson-distributed with mean μ_B, i.e. $N_B \sim \text{Pois}(\mu_B)$. Hence, μ_B determines the expected number of points in B, i.e. $\mathbb{E}\{N_B\} = \mu_B$;*

2. *for any $n \in \mathbb{N}$ and $B \subseteq \mathcal{S}$ with $0 < \mu_B < \infty$, conditionally on $N(B) = n$, then*

$$X|\{N_B = n\} \sim \text{binomial}\left(B, n, \frac{\lambda}{\mu_B}\right).$$

Then, we can write (unconditionally)

$$X \sim \text{Poisson}(\mathcal{S}, \lambda).$$

Then, we can define a Poisson process X as a random countable subset of $\mathcal{S} \subseteq \mathbb{R}^d$ where point patterns occur independently (Baddeley et al., 2006).

Now, let $N \in \mathbb{N}$ be a r.v. denoting the number of points in X. Let $\mathbf{X}_1, \cdots, \mathbf{X}_n$ be a set of n i.i.d. random vectors on $\mathcal{S}$ with density $f(\cdot) = \lambda(\cdot)/\mu$, non-negative intensity λ and intensity measure (overall intensity) given by

$$\mu = \int_{\mathcal{S}} \lambda(\mathbf{s}) d\mathbf{s}. \tag{8.3}$$

The joint conditional likelihood of $(\mathbf{X}_1 = \boldsymbol{x}_1, \cdots, \mathbf{X}_n = \boldsymbol{x}_n)$ given $N_B = n$ under an inhomogeneous Poisson process is given by (Møller and Waagepetersen, 2004)

$$f_{(\mathbf{X}_1,\cdots,\mathbf{X}_n)|N_B=n}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) = \prod_{i=1}^{n} f(\boldsymbol{x}_i) = \frac{1}{\mu^n} \prod_{i=1}^{n} \lambda(\boldsymbol{x}_i), \tag{8.4}$$

with $\boldsymbol{x}_i \in \mathbb{R}^d$. Now, let $N \sim \text{Pois}(\mu)$ for any $\mathcal{S}$ such that $\mu < \infty$. Hence, according to (8.1) and (8.4), the unconditional likelihood $f_{(N,\mathbf{X}_1,\cdots,\mathbf{X}_n)}(n, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$ is given by

$$f_{(N,\mathbf{X}_1,\cdots,\mathbf{X}_n)}(n, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) = \frac{\exp(-\mu)}{n!} \prod_{i=1}^{n} \lambda(\boldsymbol{x}_i), \tag{8.5}$$

8.3 Gaussian process modulated Cox processes

The extension of Poisson processes to stochastic intensity functions, known as doubly stochastic Poisson processes or Cox processes (Cox, 1955), enables non-parametric inference on the intensity function and allows expressing uncertainties (Møller and Waagepetersen, 2004). Previous studies have shown that other classes of point processes may also be seen as Cox processes. For example, Yannaros (1988) proved that Gamma renewal processes are Cox processes under non-increasing conditions. A similar analysis was made later for Weibull processes (Yannaros, 1994).

Definition 8.4 (Spatial Cox process) *Suppose that the intensity* $\Lambda = \{\Lambda(\boldsymbol{s}) : \boldsymbol{s} \in \mathcal{S}\}$ *is a non-negative random field such that* $\boldsymbol{s} \to \Lambda(\boldsymbol{s})$ *is a locally integrable function. If* $X|\{\Lambda = \lambda\} \sim \text{Poisson}(\mathcal{S}, \lambda)$*, then* X *is said to be a Cox process driven by* Λ*.*

Then, one can define a Cox process as a natural extension of an inhomogeneous Poisson process where λ is sampled from a non-negative stochastic process Λ.

Gaussian processes (GPs) form a flexible prior over functions, and are widely used to model the intensity process Λ (Adams et al., 2009; Donner and Opper, 2018; Fernandez et al., 2016; Gunter et al., 2014; Lasko, 2014; Lloyd et al., 2015; Møller et al., 2001; Teh and Rao, 2011). However, to ensure positive intensities, this commonly requires link functions between the intensity process and the GP Y. Typical examples of mappings are $\Lambda(x) = \exp(Y(x))$ (Diggle et al., 2013; Flaxman et al., 2015; Møller et al., 2001) or $\Lambda(x) = Y(x)^2$ (Kozachenko et al., 2016; Lloyd et al., 2015). The exponential transformation has the drawback that there is no closed-form expression for some of the integrals required to compute the likelihood. Although the square inverse link function allows closed-form expressions for certain kernels, it leads to models exhibiting "nodal lines" with zero intensity due to the non-monotonicity of the transformation (see John and Hensman, 2018, for a discussion). Furthermore, current approaches to Cox process inference cannot be used in applications such as renewal processes that require both positivity and monotonicity constraints.

Here, we introduce a novel approximation of GP-modulated Cox processes that does not rely on a mapping to obtain the intensity. In this approach we impose the constraints (e.g. non-negativeness or monotonicity) directly on Λ by sampling from a truncated Gaussian vector. This has the advantage that the likelihood in (8.5) can be computed in closed form. Moreover, the proposed approach can ensure any type of linear inequality constraint everywhere, which allows modelling of a broader range of point processes.

8.3.1 Approximation of Gaussian processes in 1D

Let $\{\Lambda(x); x \in \mathcal{S}\}$ be a centred GP on $\mathbb{R}$ with arbitrary covariance function k and compact space $\mathcal{S} = [0, 1]$. Consider a set of knots $t_1, \cdots, t_m \in \mathcal{S}$. Here we consider equispaced knots $t_j = (j-1)\Delta_m$ with $\Delta_m = 1/(m-1)$. We define Λ_m as the finite-dimensional approximation of Λ consisting of its piecewise-linear interpolation at knots

$t_1, \cdots, t_m$, i.e.,

$$\Lambda_m(x) = \sum_{j=1}^{m} \phi_j(x)\xi_j, \tag{8.6}$$

where $\xi_j := \Lambda(t_j)$ for $j = 1, \cdots, m$, and $\phi_1, \cdots, \phi_m$ are hat basis functions as in (3.2). Similarly to spline-based approaches (e.g., Sleeper and Harrington, 1990), we assume that Λ is piecewise defined by (first-order) polynomials. As discussed in Chapter 3, the striking property of this basis is that satisfying the inequality constraints (e.g. boundedness, monotonicity, convexity) at the knots implies that the constraints are satisfied everywhere in the input space (Maatouk and Bay, 2017). Although it is tempting to generalise the above construction to smoother basis functions, it makes this property difficult to enforce.

We aim at computing the distribution of Λ_m under the condition that it belongs to a convex set of functions $\mathcal{E}$ defined by some inequality constraints (e.g. positivity). As discussed in Chapter 3, the representation in (8.6) has the benefit that satisfying $\Lambda_m \in \mathcal{E}$ is equivalent to satisfying only the finite number of inequality constraints $\boldsymbol{\xi} \in \mathcal{C}$ where $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_m]^\top$, and $\mathcal{C}$ is a convex set on $\mathbb{R}^m$. For non-negativeness conditions $\mathcal{E}_+$, $\mathcal{C}$ is given by

$$\mathcal{C}_+ := \{c \in \mathbb{R}^m;\ \forall\ j = 1, \cdots, m\ :\ c_j \geq 0\}, \tag{8.7}$$

and for non-increasing conditions $\mathcal{E}_\downarrow$, $\mathcal{C}$ is given by

$$\mathcal{C}_\downarrow := \{c \in \mathbb{R}^m;\ \forall\ j = 2, \cdots, m\ :\ c_{j-1} \geq c_j\}. \tag{8.8}$$

Notice the that constraints can also be composed, e.g. the convex set of non-negativeness and non-increasing conditions is given by $\mathcal{C}_+^\downarrow = \mathcal{C}_+ \cap \mathcal{C}_\downarrow$.

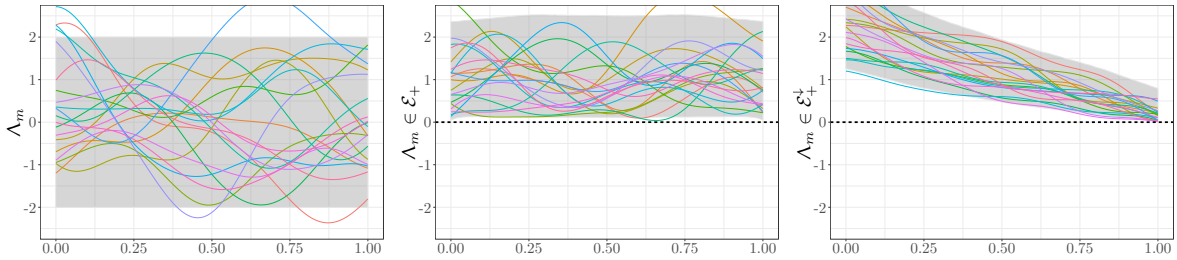
As in Chapter 3, assuming that $\boldsymbol{\xi}$ is centred Gaussian-distributed with covariance matrix $\boldsymbol{\Gamma} = (k(t_i, t_j))_{1 \leq i,j \leq m}$, then the distribution of $\boldsymbol{\xi}$ conditioned on $\boldsymbol{\xi} \in \mathcal{C}$ is a truncated Gaussian distribution, and quantifying uncertainty on Λ_m relies on sampling $\boldsymbol{\xi} \in \mathcal{C}$. We refer to Chapter 3 for a further discussion.

The effect of different inequality constraints on samples from the prior Λ_m can be seen in Figure 8.1. Here we set $m = 100$ and use a SE kernel with covariance parameters $\sigma^2 = 1$, $\ell = 0.2$ (see Table 2.1). The samples were generated via HMC (Pakman and Paninski, 2014).

8.3.2 Application to Cox processes

The key challenge in building GP-modulated Cox processes is the evaluation of the integral in the intensity measure. By considering Λ_m as the intensity of the Cox process, the intensity measure in (8.3) becomes

$$\mu_m = \int_0^1 \Lambda_m(x)dx = \int_0^1 \sum_{j=1}^{m} \phi_j(x)\xi_j dx = \sum_{j=1}^{m} c_j \xi_j,$$



(a) unconstrained GP (b) GP with $\mathcal{C}_+$ constraints (c) GP with $\mathcal{C}_+^{\downarrow}$ constraints

Figure 8.1: Samples from the prior Λ_m under (a) no constraints, (b) non-negativeness constraints, (c) both non-negativeness and non-increasing constraints. The grey region shows the 95% confidence interval.

where $c_1 = c_m = \frac{\Delta_m}{2}$ and $c_j = \Delta_m$ for $1 < j < m$. Thus, the unconditional likelihood of $(N = n, X_1 = x_1, \cdots, X_n = x_n)$ is

$$f_{(N,X_1,\cdots,X_n)|\{\xi_1,\cdots,\xi_m\}}(n, x_1, \cdots, x_n) = \frac{1}{n!} \exp\left(-\sum_{j=1}^{m} c_j \xi_j \right) \prod_{i=1}^{n} \sum_{j=1}^{m} \phi_j(x_i)\xi_j. \tag{8.9}$$

Since (8.9) depends on r.v.'s $\xi_1, \cdots, \xi_m$, it can be approximated using samples of the truncated Gaussian vector $\boldsymbol{\xi}|\{\boldsymbol{\xi} \in \mathcal{C}\}$. To estimate the covariance parameters $\boldsymbol{\theta}$ of the vector $\boldsymbol{\xi}$, one can maximise the unconditional likelihood in (8.9) via stochastic global optimisation (Jones et al., 1998).

8.3.3 Extension to high dimensions

The approximation in (8.6) can be extended to grids in d dimensions by tensorisation. For ease of notation, we assume the same number of knots m and knot-spacing Δ_m in each dimension, but the generalisation to different $m_1, \cdots, m_d$ or $\Delta_{m_1}, \cdots, \Delta_{m_d}$ is straightforward. Consider $\boldsymbol{x} = (x_1, \cdots, x_d) \in [0, 1]^d$, and a set of knots per dimension $(t_1^1, \cdots, t_m^1), \cdots, (t_1^d, \cdots, t_m^d)$. Then Λ_m is given by

$$\Lambda_m(\boldsymbol{x}) = \sum_{j_1,\cdots,j_d=1}^{m} \left[\prod_{i=1,\cdots,d} \phi_{j_i}^i(x_i) \right] \xi_{j_1,\cdots,j_d}, \tag{8.10}$$

where $\xi_{j_1,\cdots,j_d} := \Lambda(t_{j_1}, \cdots, t_{j_d})$ and $\phi_{j_i}^i$ are the hat basis functions defined in (3.2). By substituting (8.3) and (8.10), we obtain

$$\mu_m = \int_0^1 \Lambda_m(\boldsymbol{x}) d\boldsymbol{x} = \sum_{j_1,\cdots,j_d=1}^{m} \left[\prod_{i=1,\cdots,d} c_{j_i} \right] \xi_{j_1,\cdots,j_d},$$

with c_{j_i} defined as in 1D, and the likelihood is

$$f_{(N,\mathbf{X}_1,\cdots,\mathbf{X}_n)|\boldsymbol{\xi}}(n,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n) \tag{8.11}$$
$$= \frac{1}{n!}\exp\left(-\sum_{j_1,\cdots,j_d=1}^{m}\left[\prod_{i=1,\cdots,d} c_{j_i}\right]\xi_{j_1,\cdots,j_d}\right)\times\prod_{i=1}^{n}\sum_{j_1,\cdots,j_d=1}^{m}\left[\prod_{k=1,\cdots,d}\phi_{j_i}(x_{i,k})\right]\xi_{j_1,\cdots,j_d},$$

where $x_{i,k}$ is the k-th component of the point $\boldsymbol{x}_i$.

As discussed in Section 3.5, due to the tensor structure of the finite-dimensional representation, it becomes costly as the dimension d increases. The computational complexity of the HMC sampler each iteration scales with the number of inequality conditions and the number of times the HMC particles violate a constraint. This drawback could be mitigated by using sparse representations of the constraints (Pakman and Paninski, 2014), or using other types of designs of the knots (see Chapters 3, 5 and 10 for a further discussion).

8.4 Cox process inference

Having introduced the model, we now establish an inference procedure for Λ using the approximation Λ_m. For readability, we only assume non-negativeness constraints, i.e. $\boldsymbol{\xi} \geq \mathbf{0}$, but the extension to other types of constraints can be made by constructing a set of linear inequalities of the form $\boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{\xi} \leq \boldsymbol{u}$, where $\boldsymbol{A}$ is a full-rank matrix encoding the linear operations, and $\boldsymbol{l}$ and $\boldsymbol{u}$ are the lower and upper bounds (see Chapter 3). In that case, results for $\boldsymbol{A}\boldsymbol{\xi}|\{\boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{\xi} \leq \boldsymbol{u}\}$ are similar as for $\boldsymbol{\xi}|\{\mathbf{0} \leq \boldsymbol{\xi} < \boldsymbol{\infty}\}$, and samples of $\boldsymbol{\xi}$ can be recovered from samples of $\boldsymbol{A}\boldsymbol{\xi}$, by solving a linear system.

Consider the non-negative Gaussian vector $\boldsymbol{\xi}$. The posterior distribution of $\boldsymbol{\xi}$ conditioned on a point pattern $(N = n, X_1 = x_1, \cdots, X_n = x_n)$ is

$$f_{\boldsymbol{\xi}|\{N=n,X_1=x_1,\cdots,X_n=x_n\}}(\boldsymbol{\chi}) \propto f_{(N,X_1,\cdots,X_n)|\{\boldsymbol{\xi}=\boldsymbol{\chi}\}}(n,x_1,\cdots,x_n)\; f_{\boldsymbol{\xi}}(\boldsymbol{\chi}), \tag{8.12}$$

where the likelihood is defined in (8.9) and $f_{\boldsymbol{\xi}}(\boldsymbol{\chi})$ is the (truncated) Gaussian density given by

$$f_{\boldsymbol{\xi}}(\boldsymbol{\chi}) = \frac{\exp\left\{-\frac{1}{2}\boldsymbol{\chi}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{\chi}\right\}}{\int_0^\infty \exp\left\{-\frac{1}{2}\mathbf{s}^\top\boldsymbol{\Gamma}^{-1}\mathbf{s}\right\}d\mathbf{s}}, \quad \text{for} \quad \boldsymbol{\chi} \geq \mathbf{0}. \tag{8.13}$$

Since the posterior distribution (8.12) can be approximated using samples of $\boldsymbol{\xi}$, it is possible to infer Λ_m via Metropolis-Hastings.

8.4.1 Metropolis-Hastings algorithm with truncated Gaussian proposals

The implementation of the Metropolis-Hastings algorithm requires a proposal distribution q for the next step in the Markov chain. In practice, Gaussian proposals are often used, leading to the famous random-walk Metropolis algorithm (Murphy, 2012). However,

since inequality constraints are not necessarily satisfied using (non-truncated) Gaussian proposals, the standard random walk can suffer from small acceptance rates due to constraint violations. We propose as an alternative a constrained version of the random-walk Metropolis algorithm where inequality conditions are ensured when sampling from the proposal q. As $\boldsymbol{\xi}$ is (non-negative) truncated Gaussian-distributed (with prior covariance matrix $\boldsymbol{\Gamma}$), we suggest the truncated Gaussian proposal q given by

$$q(\boldsymbol{\chi}^{k+1}|\boldsymbol{\chi}^k) = \frac{\exp\left\{-\frac{1}{2}[\boldsymbol{\chi}^{k+1}-\boldsymbol{\chi}^k]^\top \boldsymbol{\Sigma}^{-1}[\boldsymbol{\chi}^{k+1}-\boldsymbol{\chi}^k]\right\}}{\int_0^\infty \exp\left\{-\frac{1}{2}[\mathbf{s}-\boldsymbol{\chi}^k]^\top \boldsymbol{\Sigma}^{-1}[\mathbf{s}-\boldsymbol{\chi}^k]\right\} d\mathbf{s}},$$

where $\boldsymbol{\chi}^{k+1}, \boldsymbol{\chi}^k \geq \mathbf{0}$ are samples of $\boldsymbol{\xi}$ and $\boldsymbol{\Sigma}$ is the covariance matrix. Sampling from q can then be performed via MCMC (Pakman and Paninski, 2014). We use $\boldsymbol{\Sigma} = \eta\boldsymbol{\Gamma}$, where η is a scale factor. This has the benefit that we are sampling from a distribution with similar structure to the true one, while η controls the step size of the Metropolis-Hastings procedure and can be manually tuned to obtain a trade-off between mixing speed and acceptance rate of the algorithm. The acceptance probability is given by

$$\alpha_k = \frac{\widetilde{f}_{\boldsymbol{\xi}|\{N=n,X_1=x_1,\cdots,X_n=x_n\}}(\boldsymbol{\chi}^{k+1})}{\widetilde{f}_{\boldsymbol{\xi}|\{N=n,X_1=x_1,\cdots,X_n=x_n\}}(\boldsymbol{\chi}^{k})} \times \beta_k, \tag{8.14}$$

where $\beta_k = q(\boldsymbol{\chi}^k|\boldsymbol{\chi}^{k+1})/q(\boldsymbol{\chi}^{k+1}|\boldsymbol{\chi}^k)$, and

$$\widetilde{f}_{\boldsymbol{\xi}|\{N=n,X_1=x_1,\cdots,X_n=x_n\}}(\boldsymbol{\chi}) = \exp\Big(-\frac{1}{2}\boldsymbol{\chi}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{\chi} - \mathbf{c}^\top\boldsymbol{\chi}\Big)\prod_{i=1}^{n}\boldsymbol{\phi}^\top(x_i)\boldsymbol{\chi} \tag{8.15}$$

is the (unnormalised) posterior distribution. $\boldsymbol{\phi}(\cdot) = [\phi_1(\cdot),\cdots,\phi_m(\cdot)]^\top$ and $\mathbf{c} = [c_1,\cdots,c_m]^\top$ are defined in (3.2) and (8.9). We now focus on the term β_k. Since the truncated Gaussian pdf has the same functional form as the non-truncated one, apart from the differing support and normalising constants, this yields

$$\beta_k = \frac{\int_0^\infty \exp\left\{-\frac{1}{2}[\mathbf{s}-\boldsymbol{\chi}^k]^\top \boldsymbol{\Sigma}^{-1}[\mathbf{s}-\boldsymbol{\chi}^k]\right\} d\mathbf{s}}{\int_0^\infty \exp\left\{-\frac{1}{2}[\mathbf{s}-\boldsymbol{\chi}^{k+1}]^\top \boldsymbol{\Sigma}^{-1}[\mathbf{s}-\boldsymbol{\chi}^{k+1}]\right\} d\mathbf{s}}. \tag{8.16}$$

The orthants $\int_0^\infty \exp\left\{-\frac{1}{2}[\boldsymbol{x}-\boldsymbol{\mu}]^\top \boldsymbol{\Sigma}^{-1}[\boldsymbol{x}-\boldsymbol{\mu}]\right\} d\boldsymbol{x}$ cannot be computed in closed form, but they can be estimated via MC (Botev, 2017; Genz, 1992). Algorithm 3 summarises the implementation of the Metropolis-Hastings algorithm for the Cox process inference using the proposed finite-dimensional approximation.

8.4.2 Inference with multiple observations

For N_o independent observations $(X_{\nu,1},\cdots,X_{\nu,n_\nu})$ with $\nu = 1,\cdots,N_o$, the acceptance probability follows

$$\alpha_k = \frac{\prod_{\nu=1}^{N_o} f_{\boldsymbol{\xi}|\{N_\nu=n_\nu,X_{\nu,1}=x_{\nu,1},\cdots,X_{\nu,n_\nu}=x_{\nu,n_\nu}\}}(\boldsymbol{\chi}^{k+1})}{\prod_{\nu=1}^{N_o} f_{\boldsymbol{\xi}|\{N_\nu=n_\nu,X_{\nu,1}=x_{\nu,1},\cdots,X_{\nu,n_\nu}=x_{\nu,n_\nu}\}}(\boldsymbol{\chi}^{k})}\,\beta_k, \tag{8.17}$$

with posterior $f_{\boldsymbol{\xi}|\{N_\nu=n_\nu,X_{\nu,1}=x_{\nu,1},\cdots,X_{\nu,n_\nu}=x_{\nu,n_\nu}\}}$ and β_k given by (8.12) and (8.16). Then, Algorithm 3 can be used with (8.17).

Algorithm 3 Metropolis-Hastings algorithm for Cox process inference with truncated Gaussian proposals

1: Input: $\boldsymbol{\chi}^{(0)} \in (\mathbb{R}^m)^+$, $\boldsymbol{\Gamma}$ (covariance matrix of $\boldsymbol{\xi}$), η (scale factor).
2: **for** $k = 0, 1, 2, \cdots$ **do**
3: Sample $\boldsymbol{\chi}' \sim \mathcal{N}\left(\boldsymbol{\chi}^{(k)}, \eta\boldsymbol{\Gamma}\right)$ such that $\boldsymbol{\chi}' \in \mathcal{C}_+$.
4: Compute α_k as in (8.14).
5: Sample $u_k \sim \text{uniform}(0, 1)$.
6: Set new sample to
7: $$\boldsymbol{\chi}^{(k+1)} = \begin{cases} \boldsymbol{\chi}', & \text{if } \alpha_k \geq u_k \\ \boldsymbol{\chi}^{(k)}, & \text{if } \alpha_k < u_k \end{cases}.$$
8: Compute $\lambda_m^{(k)}(x) = \sum_{j=1}^m \phi_j(x)\chi_j^{(k)}$ at location x with ϕ_j defined in (3.2).

8.4.3 Numerical illustrations

8.4.3.1 Illustrations on 1D

We test the performance of the finite approximation of GP-modulated Cox process on 1D and 2D applications. In the following, we use the squared-exponential covariance for the Gaussian vector $\boldsymbol{\xi}$ so that we can compare to Lloyd et al. (2015). We estimate the covariance parameters $\boldsymbol{\theta} = (\sigma^2, \ell)$ by maximising the likelihood in (8.9). For all numerical experiments, we fix m such that we obtain accurate resolutions of the finite representations while minimising the cost of MCMC (see Bay et al., 2016; Maatouk and Bay, 2017, for discussion about the convergence of the finite-dimensional approximation of GPs).[1] For sampling $\boldsymbol{\xi}$, we use the HMC sampler proposed by Pakman and Paninski (2014). To approximate the Gaussian orthant probabilities from (8.16), we use the estimator proposed by Botev (2017) using 200 MC samples. We run Algorithm 3 with a scale factor η between 10^{-3} and 10^{-4} for a good trade-off between the mixing speed and the acceptance rate for each experiment. The number of discarded burn-in samples until the Markov chains became stationary varied between 10^3 and 10^4 samples. Codes were implemented based on the R package `lineqGPR` (López-Lopera, 2019).

Here, we test the approach using the toy examples proposed by Adams et al. (2009):

$$\lambda_1(x) = 2\exp\{-x/15\} + \exp\{-[(x-25)/10]^2\},$$
$$\lambda_2(x) = 5\sin(x^2) + 6,$$
$$\lambda_3(x) = \text{piecewise linear through } (0,2), (25,3), (50,1), (75,2.5) \text{ and } (100,3).$$

The domains for λ_1, λ_2 and λ_3 are $\mathcal{S}_1 = [0, 50]$, $\mathcal{S}_2 = [0, 5]$ and $\mathcal{S}_3 = [0, 100]$, respectively.

Figure 8.2 shows the inference results using $N_o = 1, 10, 100$ observations sampled from the ground truth. With increasing number of observations the inferred intensity converges to the ground truth. Here, we fixed $m = 100$ and $\eta = 10^{-3}$.

[1] We tested the model for various values of m, observing that, after a certain value, inference results are unchanged. As a rule of thumb, the number of knots per dimension can be set to $m_i = 10 \cdot \text{range}(\mathcal{S}_i)/\ell_i$ for $i = 1, \cdots, d$.

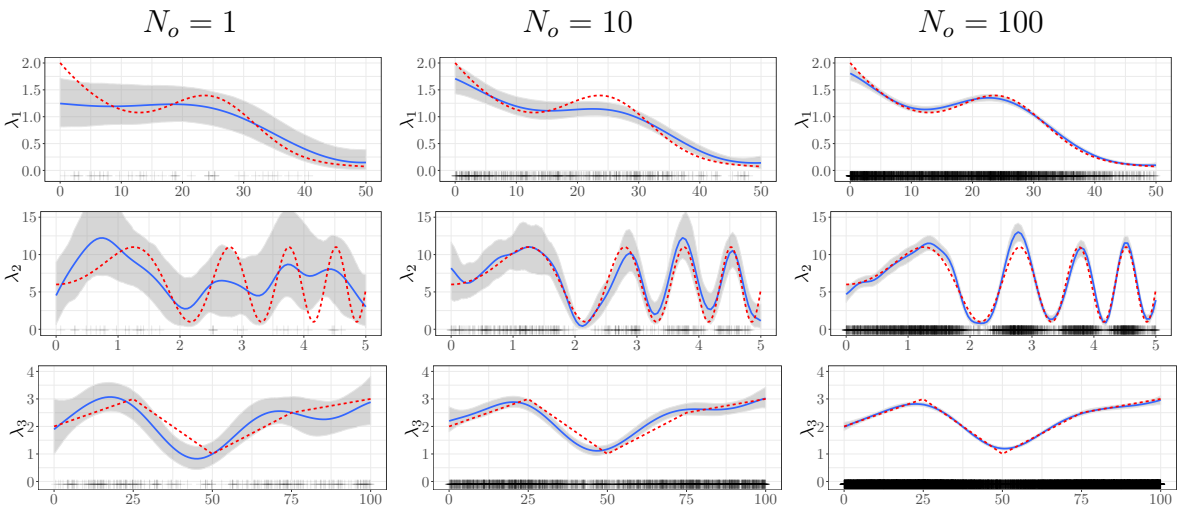


Figure 8.2: Inference results with multiple observations ($N_o = 1, 10, 100$) using the toy examples from (Adams et al., 2009). Each panel shows the point patterns (crosses), the true intensity λ (dashed lines) and the intensity inferred by the finite approximation of GP-modulated Cox processes (solid lines). The estimated 90% confidence intervals of the finite approximation are shown in grey.

In Table 8.1, we assess the performance of the proposed GP-based approach under non-negativeness constraints (cGP-c_+). We compare our inference results to the ones obtained with a log-Gaussian process (log-GP) modulated Cox process (Møller et al., 2001) and Variational Bayes for Point Processes (VBPP) (Lloyd et al., 2015) using the Q^2 criterion in (3.17). Here, Q^2 is equal to one if the inferred $\widehat{\lambda}$ is exactly equal to the true λ, zero if it is equal to the average intensity $\overline{\lambda}$, and negative if it performs worse than $\overline{\lambda}$. We compute the Q^2 indicator on a regular grid of 1000 locations in $\mathcal{S}$. Then, we compute the mean μ and one standard deviation σ of the Q^2 results across 20 different replicates. Table 8.1 shows that our approach outperforms its competitors, with consistently higher means of the Q^2 results and lesser dispersion σ.

We assess the computational cost of the proposed approach using the third toy example λ_3 for $N_o = 100$ (which has the largest number of events with on average 22500 events in total). Obtaining one sample using our approach takes around 60 milliseconds, and generating all 10^4 samples takes 10 minutes in total (in contrast to the 18 minutes required by VBPP).[2] The multivariate effective sample size (ESS) (Flegal et al., 2017) was estimated at 322, corresponding to an effective sampling rate of 0.536 s^{-1}.

8.4.3.2 Illustrations on 2D

In this toy example, we test the framework on a 2D spatial toy example. We sample 2D random point patterns from three Gaussian distributions with same covariance matrix

[2]These experiments were executed on a single core of an Intel® Core™ i7-6700HQ CPU.

Table 8.1: Q^2 results for the toy examples in Figure 8.2, averaged over 20 (†10) replicates. Our results (cGP-c_+) are compared to results for Møller et al. (2001) (log-GP) and Lloyd et al. (2015) (VBPP).

Toy	N_o	Q^2 ($\mu \pm \sigma$) [%] log-GP	VBPP	cGP-c_+
λ_1	1	51.2±30.1	51.9±26.1	**65.7±14.3**
	10	95.1± 3.9	94.6± 3.7	**95.4± 2.3**
	100	**99.5± 0.2**	**99.5± 0.3**	**99.5± 0.3**
λ_2	1	-35.2±43.4	-1.1±28.8	**0.7±24.0**
	10	72.6± 9.1	71.7±10.4	**81.9± 7.4**
	100	95.4± 0.7	92.1± 3.9	**97.8± 0.6**
λ_3	1	49.2±22.6	49.5±29.9	**58.1±21.4**
	10	91.7± 4.4	93.8± 2.8	**94.3± 2.5**
	100	98.4± 0.4	**98.9±0.3**†	**98.8± 0.3**

but at different locations on $\mathcal{S} \in [0,1]^2$. The centres of the distributions are placed at $(0.2, 0.2)$, $(0.2, 0.8)$ and $(0.8, 0.6)$. We used isotropic 2D SE kernels as covariance functions, with the same variance parameter $\sigma^2 = 1 \times 10^{-3}$. We randomly simulate 10 events from each Gaussian distribution for a total of 30 (spatial) point patterns.

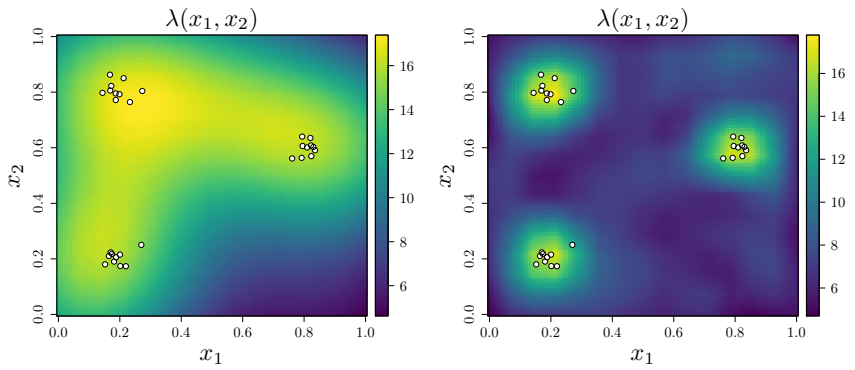
Figure 8.3 shows the performance of the proposed framework with different length-scale parameters (ℓ_1, ℓ_2). One can observe the capability of the model to find a trade-off between regularity and fidelity by properly controlling the values of the length-scales.

8.5 Applications

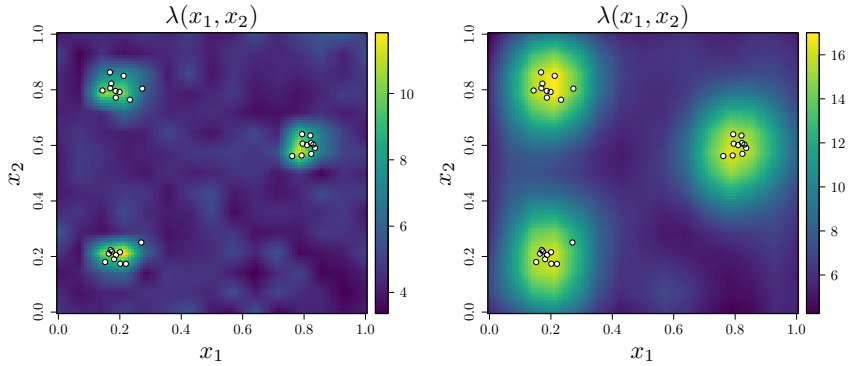
8.5.1 Renewal point processes

Poisson processes have been extended to model renewal processes where intensity functions are seen as hazard rates defining the probability that an operating object fails (Cha and Finkelstein, 2018; Serfozo, 2009). However, in many application, e.g. reliability engineering and survival analysis, hazard rates exhibit monotonic behaviours describing the degradation of items or lifetime of organisms. For example, the hazard functions for the failure of many mechanistic devices and the mortality of adult humans tend to exhibit monotonic behaviours. Thus, taking monotonicity constraints into account in renewal processes is crucial for the study of many applications. Moreover, it is known that introducing monotonicity information in GPs can lead to more realistic uncertainties (Maatouk and Bay, 2017; Riihimäki and Vehtari, 2010).

As discussed in Section 8.3, some renewal processes can be seen as Cox processes under certain conditions. In order to demonstrate that we can model other types of point



(a) $\ell_1 = \ell_2 = 3 \times 10^{-1}$ (b) $\ell_1 = \ell_2 = 10^{-1}$



(c) $\ell_1 = \ell_2 = 10^{-2}$ (d) $\widehat{\ell}_1 = 0.11, \widehat{\ell}_2 = 0.17$

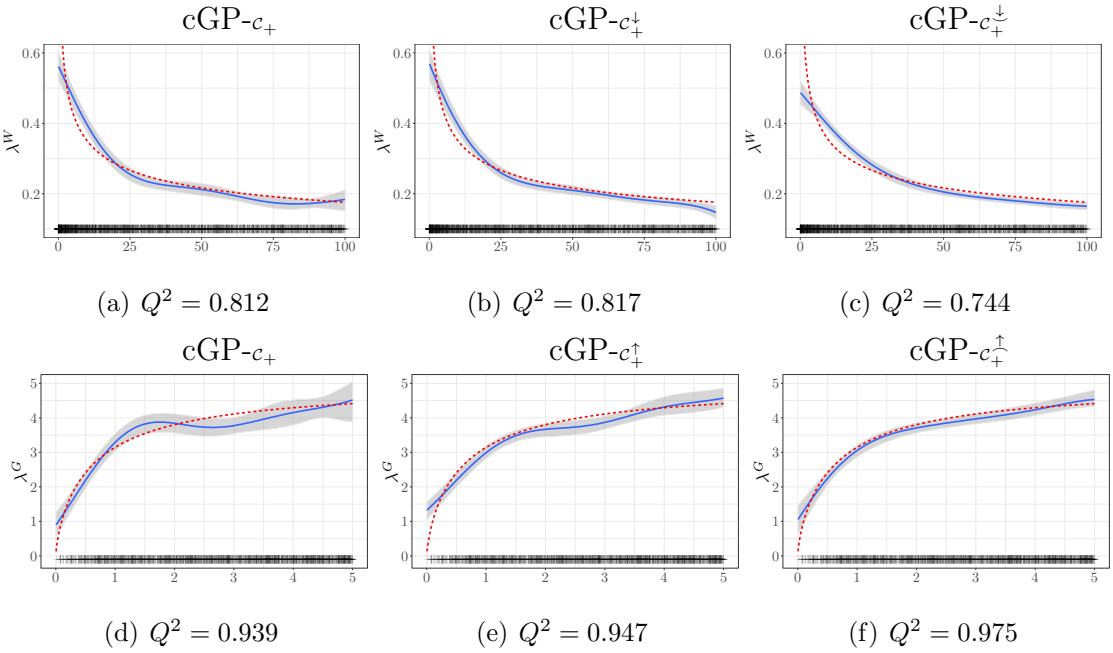
Figure 8.3: Inference results on a 2D spatial toy example with different length-scale parameters (ℓ_1, ℓ_2). Each panel shows the point patterns (white dots) and the intensity inferred by the finite approximation of GP-modulated Cox processes.

patterns, here we use two toy examples where hazard rates are known to be monotonic. Both examples are inspired by two classical renewal processes: Weibull process and Gamma process.

For the first class, the Weibull hazard function is

$$\lambda^W(x) = \alpha\beta x^{\beta-1} \quad \text{for} \quad x \geq 0, \tag{8.18}$$

where α and β are the scale parameter and shape parameter, respectively. Depending on β, λ^W can be either non-increasing $(0 < \beta < 1)$, constant $(\beta = 1)$, or non-decreasing $(\beta > 1)$. Moreover, for $\beta \in (0, 1]$, the Weibull renewal process can be seen as a Cox process (Yannaros, 1988). For numerical experiments, we consider the case of non-increasing conditions in the domain $\mathcal{S} = [0, 100]$ by fixing $\alpha = 1$ and $\beta = 0.7$ (see Figure 8.4). We test the framework using $N_o = 100$ observations from λ^W, and we consider non-negativeness conditions, with (cGP-$c_+^{\downarrow}$) or without (cGP-c_+) taking into account the non-increasing constraint. We also consider the case where λ^W is non-increasing and convex (cGP-$c_+^{\downarrow\downarrow}$).



(a) $Q^2 = 0.812$ (b) $Q^2 = 0.817$ (c) $Q^2 = 0.744$

(d) $Q^2 = 0.939$ (e) $Q^2 = 0.947$ (f) $Q^2 = 0.975$

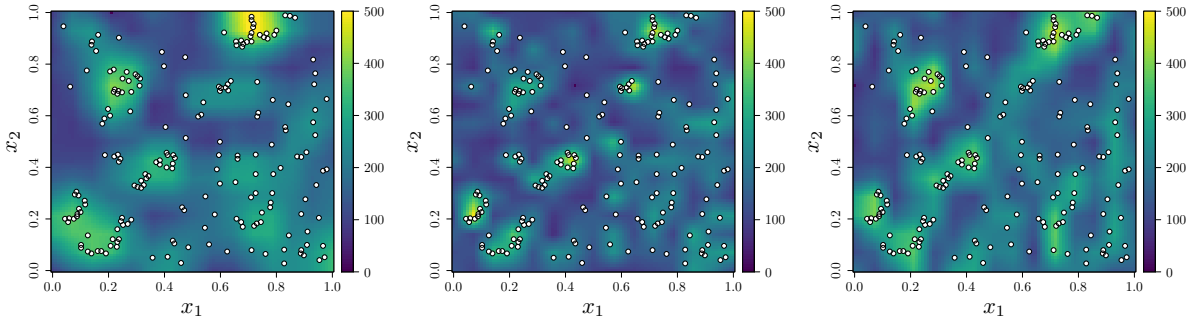
Figure 8.4: Renewal inference examples under different inequality constraints using $N_o = 100$ and $m = 100$. Inference results are shown for (top) a Weibull renewal process with $\alpha = 1$ and $\beta = 0.7$, and (bottom) a Gamma renewal process with $\alpha = 5$ and $\beta = 1.7$. The panel description is the same as in Figure 8.2.

For the Gamma class, the hazard function is given by

$$\lambda^G(x) = \frac{\alpha\, x^{\beta-1} e^{-x}}{\Gamma(\beta) - \Gamma_x(\beta)}, \quad \text{for} \quad x \geq 0, \tag{8.19}$$

where Γ and Γ_x are the Gamma function and the incomplete Gamma function, respectively (Cha and Finkelstein, 2018), and α and β are the scale parameter and shape parameter. As for the Weibull process, different behaviours can be obtained using different values of β. Since similar profiles are obtained for $\beta \in (0, 1]$, here we are interested in the case where λ^G exhibits non-decreasing constraints ($\beta > 1$). We fix $\mathcal{S} = [0, 5]$, $\alpha = 5$ and $\beta = 1.7$ obtaining a non-decreasing profile as shown in Figure 8.4. Here, we consider non-decreasing (cGP-$c_+^{\uparrow}$), and non-decreasing and concave (cGP-$c_+^{\uparrow\uparrow}$) constraints. Since $\lambda^G(x) < \alpha$ for $x \in \mathcal{S}$, we add the constraint $\lambda^G \in [0, \alpha]$.

Figure 8.4 shows the inferred intensities of λ^W and λ^G under the different conditions previously discussed. In both experiments, we fixed $m = 100$ and $\eta = 10^{-4}$. For the Weibull class λ^W, the performance of all three models, cGP-c_+, cGP-$c_+^{\downarrow}$ and cGP-$c_+^{\downarrow\downarrow}$, tends to be similar. However, the model without monotonicity constraint exhibits undesired oscillations, whereas the other two approaches provide more realistic decreasing profiles and more accurate inference results for $x > 50$. We can also observe that the three models cannot learn the singularity at $x = 0$. Note that the proposed methodology does not make any assumption on the kernel, and it would be possible to consider a covariance



(a) $\ell_1 = \ell_2 = 0.1$ (b) $\ell_1 = \ell_2 = 0.01$ (c) $\widehat{\ell}_1 = 0.055$, $\widehat{\ell}_2 = 0.084$

Figure 8.5: Inference results of the redwoods data in (Baddeley et al., 2015; Ripley, 1977). Each panel shows the point patterns (white dots) and the estimated intensity λ.

function such as $k(x, y)/(xy)$ in order to improve the model behaviour for small and large values of x. For the Gamma hazard function λ^G, one may clearly observe the benefits of adding the non-decreasing and concave constraints, obtaining absolute improvements between 0.8% and 3.6% of the Q^2 indicator. Both examples of Figure 8.4 show that the monotonicity and convexity conditions found in certain point processes can be difficult to learn directly from the data. This suggests that including those constraints in the GP prior is necessary to get accurate models with more realistic uncertainties.

8.5.2 Locations of redwood trees

We now assess the performance of the proposed approach for a 2D spatial problem. We use the dataset provided by Ripley (1977) which describes the locations of redwood trees. The dataset contains $n = 195$ events scaled to the unit square (see Figure 8.5). Here we choose $m = 15$, obtaining 225 knots in total, to obtain a good trade-off between resolution and computational cost. We use the product of two SE kernels with covariance parameters $\boldsymbol{\theta} = (\sigma^2, \ell_1, \ell_2)$ as the covariance function of the Gaussian vector $\boldsymbol{\xi}$, and we choose $\eta = 10^{-4}$ in Algorithm 3. Following the burn-in step, we keep 10^5 samples for the inference of λ, yielding a total running time of 7.6 hours (i.e. a sampling rate of approximately 4 s^{-1}).

Figure 8.5 shows the normalised inference results for the redwood dataset for different values of the length-scale parameters. Since in the proposed approach we directly impose the inequality conditions on the Gaussian vector $\boldsymbol{\xi}$ instead of using a link function, the interpretation of the length-scale parameters (ℓ_1, ℓ_2) are the same as for standard GPs: one can find a trade-off between fidelity and regularity by tuning ℓ. One can note, from Figures 8.5(a) and 8.5(b), that both profiles tend to properly learn the point patterns but more regularity is exhibited when $\ell_1 = \ell_2 = 10^{-1}$. For the case $\ell_1 = \ell_2 = 10^{-2}$, although the model follows the point patterns, one may observe noisy behaviour in regions without points, e.g. around $(x_1, x_2) = (0.30, 0.85)$, as small values of ℓ lead to more oscillatory Gaussian random fields. Finally, we infer λ when the covariance parameters $\boldsymbol{\theta}$

are estimated via maximum likelihood using (8.11). According to the estimated length-scales ($\widehat{\ell_1} = 0.055$, $\widehat{\ell_2} = 0.084$), one can conclude that the estimated intensity λ is smoother along the second dimension x_2. This is in agreement with the inference results by Adams et al. (2009), where more variations of λ were exhibited across x_1.

8.6 Conclusions

The proposed model for GP-modulated Cox processes is based on a finite-dimensional approximation of a GP that is constrained to be positive. This approach shows several advantages. First of all, it is based on general linear inequality constraints so it allows us to incorporate more information, such as monotonicity and convexity, in the prior. As seen in the experiments, this appears to be particularly helpful when few data are available. Second, imposing directly the positivity constraint on the GP makes the use of a link function unnecessary. Both the likelihood and the intensity measure can be computed analytically, which is not always the case when using a link function. Finally, the fact that the proposed model is based on a finite-dimensional representation ensures that the computational burden grows linearly with the number of observations.

There are two key elements that make the method work: (a) the finite-dimensional representation of the GP that ensures that the constraints are satisfied everywhere, and (b) the dedicated MCMC proposal distribution based on a truncated normal distribution which allows us to have high acceptance rates compared to a naive multivariate Gaussian proposal.

The main limitation regarding the scaling of the proposed method lies in the dimension of the input space. This is due to the construction by tensorisation of the basis functions used to obtain the finite-dimensional representation. Moreover, the proposed model is also sensitive to three parameters: the dimensionality of the space in which we perform HMC, the number of constraints, and the number of times the HMC particles violate a constraint. However, we believe that these limitations are not inherent to the proposed model and that other types of designs of the knots (e.g. sparse designs) could be used in high dimensions.

Part VII

LineqGPR: An R Package for Gaussian Process Regression models with Linear Inequality Constraints

Chapter 9

lineqGPR Package (v.0.0.4)

Contents

9.1 Introduction

As shown in previous chapters, introducing inequality constraints in GP models can lead to more accurate uncertainty quantification in a great variety of applications. Based mainly on the framework detailed in Chapter 3, `lineqGPR` is an R[1] package for GP regression modelling with inequality constraints (López-Lopera, 2019). It can be used to create GP models under boundedness, monotonicity, convexity constraints or a combination of these. Furthermore, it also allows users to design their own sets of linear inequalities (see Chapter 3). `lineqGPR` contains all the typical features of classic GP libraries, e.g. the parameter estimation via maximum likelihood (see Chapters 2 and 7), support for noisy observations (see Chapter 4). Finally, recent developments allow the implementation of additive GP models under linear inequality constraints, leading to models that can be used for thousands of observations and for high dimensions (see Chapter 6).

`lineqGPR` is available on CRAN[2] as an open-source software licensed under GNU GPL-3. It is based on previous R packages produced by the Deep Inside Computer Experiments (Dice) and ReDice Consortiums (e.g. `DiceKriging`, Roustant et al. (2012); `DiceDesign`, Dupuy et al. (2015); `kergp`, Deville et al. (2015)), but incorporating structures of classic

[1] R is a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques (R Core Team, 2018).

[2] The Comprehensive R Archive Network (CRAN) is a network of file transfer protocol (FTP) and web servers around the world that store identical, up-to-date, versions of code and documentation for R.

GP libraries from other platforms (e.g. the GPmat toolbox from MATLAB, and the GPy library from Python).

The main functionalities of `lineqGPR` are implemented as S3 methods.

Method Name	Description
create	Creation function of GP models under linear inequality constraints.
lineqGPOptim	Covariance parameter estimation under linear inequality constraints.
predict	Prediction of the objective function at new points using a GP model.
simulate	Simulation of GP models under linear inequality constraints.
plot, ggplot	Plot for a constrained GP models.

It also contains implementations of various MC and MCMC samplers for the approximation of truncated multinormals. The samplers are based on recent contributions on efficient inference methods: rejection sampling from the mode (`RSM`, Maatouk and Bay, 2016), rejection sampling via exponential tilting (`ExpT`, Botev, 2017), Gibbs sampling (`Gibbs`, Taylor and Benjamini, 2016), and Hamiltonian Monte Carlo (`HMC`, Pakman and Paninski, 2014).

We refer to the documentation of the `lineqGPR` package available on CRAN for further details on additional functions, utilities and demos:

- López-Lopera, A. F. (2019). lineqGPR: Gaussian Process Regression Models with Inequality Constraints. R package version 0.0.4. Link: https://cran.r-project.org/web/packages/lineqGPR/.

9.2 Demos

Here, we illustrate how `lineqGPR` works on 1D and 2D examples. The codes are based on a Jupyter notebook[3] provided by the Chair OQUAIDO in order to reproduce results from previous chapters. They were executed on a single core of an Intel® Core™ i7-6700HQ CPU. We refer to the instruction manual in (López-Lopera, 2019) for further examples:

1. See: `help("lineqGPR-package")`.
2. See: `demo(package="lineqGPR")`.
3. See also the examples of the inner functions of the package.

```
In [1]:
    library(lineqGPR)  # package GPs under inequality constraints
    library(ggplot2)   # package for graphics and visualisation
```

[3] Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualisations and narrative text.

9.2.1 1D Gaussian processes with inequality constraints

First, we aim at reproducing similar 1D toy examples as the ones in Figures 3.1 and 3.2. In that case, we use the model in (3.1) under different types of inequality constraints.

Let the target Sigmoid function be given by

$$x \mapsto \frac{1}{1+\exp(-7[x-0.5])}, \quad \text{for} \quad x \in [0,1]. \tag{9.1}$$

Now, we implement GP models with different types of constraints: either without constraints, or with boundedness or monotonicity constraints. We fixed the number of knots $m = 100$, and we simulated samples via HMC (see subsection 3.3.2). In that case, we need first to pass some arguments to create the model:

- The class of the model: `"lineqGP"`.
- The interpolation points: (x_i, y_i) for $i = 1, \cdots, n$.
- The type of constraints: e.g. `"boundedness"`, `"monotonicity"` or `"convexity"`. If `constrType == "boundedness"`, we can define the bounds: `bounds <- c(0,1)`.

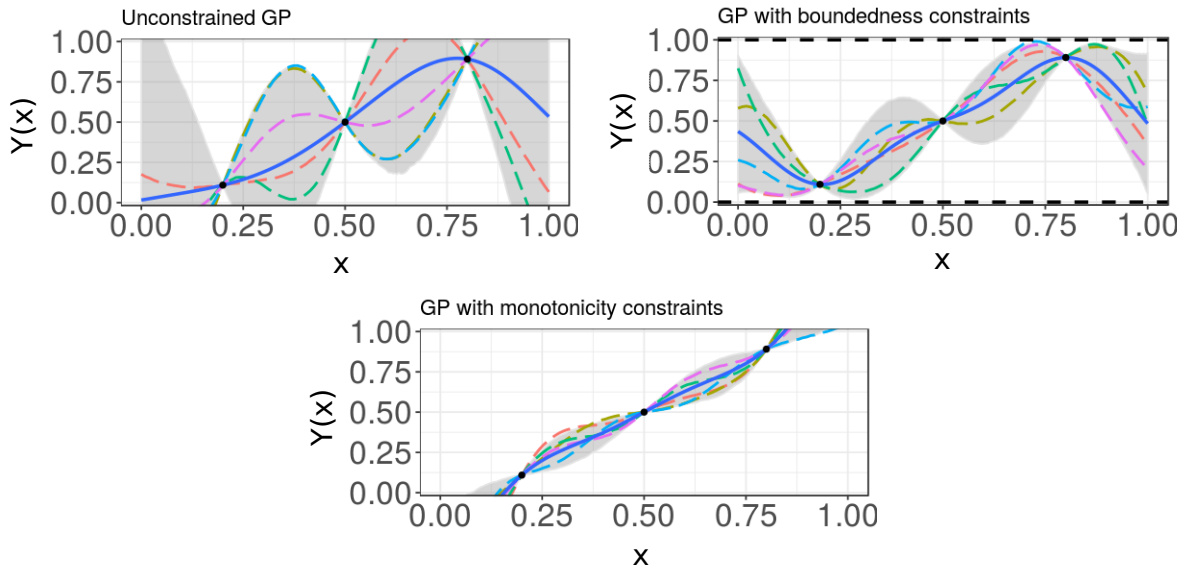
After creating the model, one can then modify the default parameters:

- The type of sampler (`HMC` by default): `model$localParam$sampler <- "HMC"`.
- The number of knots m from the object: `model$localParam$m <- 100`.

In this cartoon example, we evaluated (9.1) at $x_1 = 0.2$, $x_2 = 0.5$ and $x_3 = 0.8$, and we used those evaluations as a DoE for the interpolation conditions.

```
In [3]: #### Generating the synthetic dataset ####
    sigfun <- function(x) return(1/(1+exp(-7*(x-0.5))))
    x <- seq(0, 1, 0.001); y <- sigfun(x)
    DoE <- splitDoE(x, y, DoE.idx = c(201, 501, 801))

    #### GP with nearly inactive boundedness constraints [-10,10] ####
    # creating the "lineqGP" model
    model <- create(class = "lineqGP", x = DoE$xdesign, y = DoE$ydesign,
                    constrType = "boundedness")
    model$localParam$m <- 100 # changing the (default) number of knots
    model$bounds <- c(-10,10) # changing the (default) bounds
    # sampling from the model
    sim.model <- simulate(model, nsim = 1e3, seed = 1, xtest = DoE$xtest)
    ggplotLineqGPModel <- ggplot(sim.model)

    #### GP with active boundedness constraints [0,1] ####
    model <- create(class = "lineqGP", x = DoE$xdesign, y = DoE$ydesign,
                    constrType = "boundedness")
    model$localParam$m <- 100 # changing the (default) number of knots
    model$bounds <- c(0,1) # changing the (default) bounds
```

```
# sampling from the model
sim.model <- simulate(model, nsim = 1e3, seed = 1, xtest = DoE$xtest)
ggplotLineqGPModel <- ggplot(sim.model)

#### GP with monotonicity constraints ####
model <- create(class = "lineqGP", x = DoE$xdesign, y = DoE$ydesign,
                constrType = "monotonicity")
model$localParam$m <- 100 # changing the (default) number of knots
# sampling from the model
sim.model <- simulate(model, nsim = 1e2, seed = 1, xtest = DoE$xtest)
ggplotLineqGPModel <- ggplot(sim.model)
```

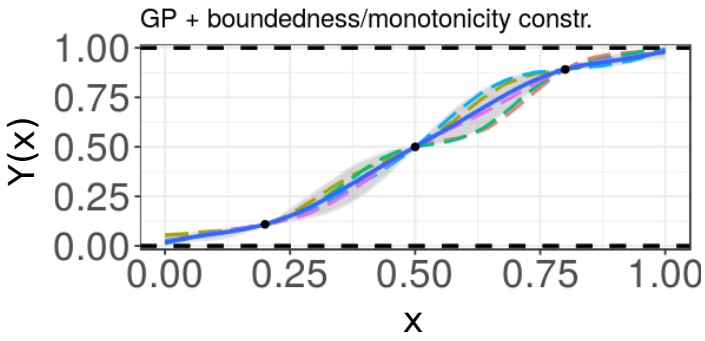


One can also impose multiple constraints by concatenating predefined ones, e.g.,

- `model$constrType = c("boundedness", "monotonicity")`

Next, we show an example of a GP model with both boundedness and monotonicity constraints for the Sigmoid example in (9.1).

```
In [4]: #### GP with both boundedness and monotonicity constraints ####
    # creating the "lineqGP" model
    model <- create(class = "lineqGP", x = DoE$xdesign, y = DoE$ydesign,
                    constrType = c("boundedness", "monotonicity"))
    model$localParam$m <- 100  # changing the (default) number of knots
    # modifying bounds for first arg of "constrType" (boundedness)
    model$bounds[1,] <- c(0,1)

    # sampling from the model
    sim.model <- simulate(model, nsim = 1e2, seed = 1, xtest = DoE$xtest)
    ggplotLineqGPModel <- ggplot(sim.model, bounds = c(model$bounds[1,]))
```



Moreover, we can design our own linear inequality constraints by building the set of linear inequalities

$$\boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u},$$

where $\boldsymbol{\Lambda}$ encodes the linear operations between the knots, the $\mathbf{l}$ and $\mathbf{u}$ represent the lower and upper bounds (respectively). We refer to Section 3.2 for a further discussion.

Now, we show an example where we aim at imposing monotonicity constraint in the first 50 knots, and boundedness constraints $Y_m \in [0.5, 1]$ over the last 50 knots. Then, the set of inequalities are given by

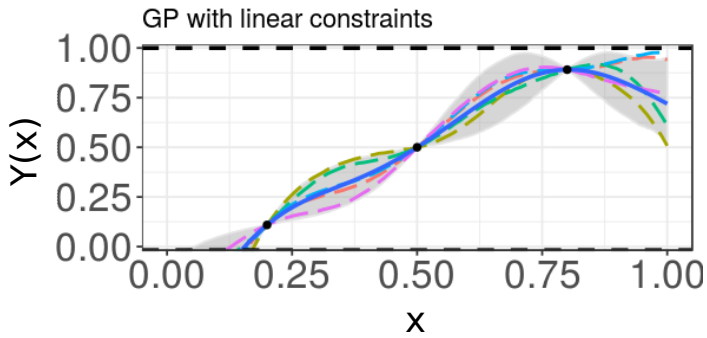
$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_M & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_B \end{bmatrix}, \qquad \boldsymbol{l} = \begin{bmatrix} \boldsymbol{l}_M \\ \boldsymbol{l}_B \end{bmatrix}, \qquad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_M \\ \boldsymbol{u}_B \end{bmatrix},$$

where

$$\underbrace{\begin{bmatrix} -\infty \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\boldsymbol{l}_M} \leq \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 1 \end{bmatrix}}_{\boldsymbol{\Lambda}_M} \leq \underbrace{\begin{bmatrix} \infty \\ \infty \\ \vdots \\ \infty \end{bmatrix}}_{\boldsymbol{u}_M}, \quad \text{and} \quad \underbrace{\begin{bmatrix} 0.5 \\ 0.5 \\ \vdots \\ 0.5 \end{bmatrix}}_{\boldsymbol{l}_B} \leq \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}}_{\boldsymbol{\Lambda}_B} \leq \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{\boldsymbol{u}_B}.$$

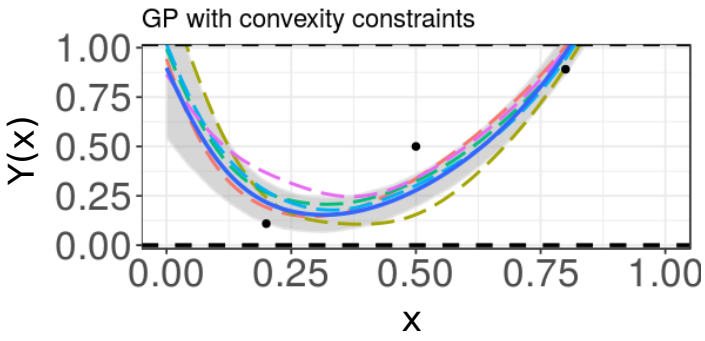
```
In [5]: #### GP with linear constraints ####
    # creating the "lineqGP" model
    model <- create(class = "lineqGP", x = DoE$xdesign, y = DoE$ydesign,
                    constrType = "linear")
    m <- model$localParam$m <- 100 # changing the number of knots

    ## building the linear inequality constraints
    # imposing monotonicity constraints for 50 knots
    lsys1 <- lineqGPSys(m = 50, constrType = "monotonicity",
                        l = 0, u = Inf, rmInf = FALSE) # rm inactive constr.
    # imposing boundedness constraints Y_m in (0.5,1) for 50 knots
    lsys2 <- lineqGPSys(m = 50, constrType = "boundedness",
                        l = 0.5, u = 1, rmInf = FALSE) # rm inactive constr.
```

```
    # stacking the constraints such that monotonocity conditions are imposed
    # for the first 50 knots and boundedness conditions for the last 50 knots
    A <- matrix(0, m, m)
    A[1:(m/2), 1:(m/2)] <- lsys1$A
    A[(m/2+1):m, (m/2+1):m] <- lsys2$A
    # passing the inequalities to the model
    model$Lambda <- A
    model$lb <- c(lsys1$l, lsys2$l)
    model$ub <- c(lsys1$u, lsys2$u)

    # sampling from the model
    sim.model <- simulate(model, nsim = 1e3, seed = 1, xtest = DoE$xtest)
    # plotting samples
    ggplotLineqGPModel <- ggplot(sim.model, bounds = c(-Inf, 1)) +
         geom_hline(yintercept = c(0,1), linetype = "dashed")
```



Finally, one must note that the constraints can be imposed if and only if they are satisfied at the interpolation points y_i for $i = 1, \cdots, n$. However, one can add a noise effect in order to relax the interpolation constraints: `model$varnoise`. For the Sigmoid example, one can observe that the interpolation points do not meet convexity constraints. We then introduce a noise term in the GP model for the relaxation of the interpolation constraints letting the GP trajectories to be convex.

```
In [6]: #### GP with convexity constraints ####
    # creating the "lineqGP" model
    model <- create(class = "lineqGP", x = DoE$xdesign, y = DoE$ydesign,
                    constrType = "convexity")
    model$localParam$m <- 100 # changing the (default) number of knots
    model$varnoise <- 5e-3 # adding the noise variance

    # sampling from the model
    sim.model <- simulate(model, nsim = 1e3, seed = 1, xtest = DoE$xtest)
    ggplotLineqGPModel <- ggplot(sim.model, bounds = c(model$bounds[1,]))
```

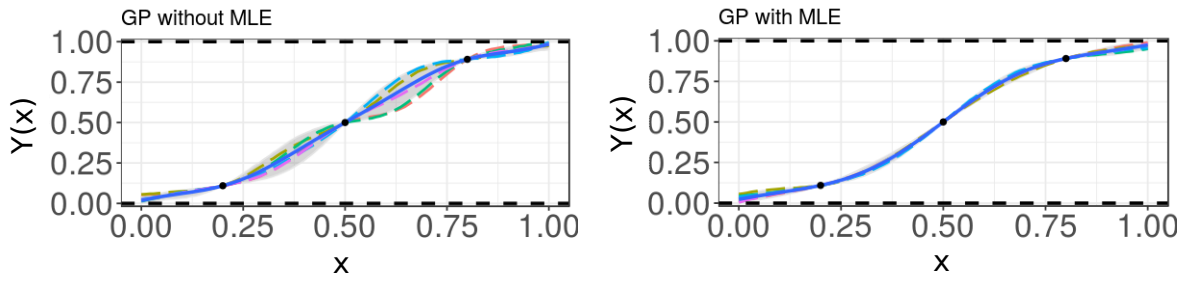


9.2.2 Covariance parameter estimation

As discussed in Chapters 2 and 7, the covariance parameters are usually estimated via ML. This can be achieved by the `lineqGPOptim` function. Continuing with the Sigmoid example proposed above, we estimate the covariance parameters of the constrained GP model via (unconstrained) ML.

```
In [7]: set.seed(7)
    #### GP with both boundedness and monotonicity constraints ####
    model <- create(class = "lineqGP", x = DoE$xdesign, y = DoE$ydesign,
                    constrType = c("boundedness", "monotonicity"))
    model$localParam$m <- 100 # changing the (default) number of knots
    # modifying the bounds for first arg of "constrType" (boundedness)
    model$bounds[1, ] <- c(0,1)

    # sampling from the model
    sim.model <- simulate(model, nsim = 1e2, seed = 1, xtest = DoE$xtest)
    message("Initial covariance parameters: ", model$kernParam$par[1],
            ", ", model$kernParam$par[2])
    ggplotLineqGPModel <- ggplot(sim.model, bounds = c(model$bounds[1,]))

    # estimating the covariance parameter via MLE
    model2 <- lineqGPOptim(model,
                           opts = list(algorithm = "NLOPT_LD_MMA",
                                       print_level = 0,
                                       ftol_abs = 1e-3, maxeval = 20,
                                       check_derivatives = TRUE,
                                       parfixed = c(FALSE, FALSE)),
                           lb = c(0.1, 0.01), ub = c(2,  0.3))
    message("Estimated covariance parameters via MLE: ",
            model2$kernParam$par[1], ", ", model2$kernParam$par[2])

    # evaluating the "optimal" model
    sim.model2 <- simulate(model2, nsim = 1e2, seed = 1, xtest = DoE$xtest)
    ggplotLineqGPModel <- ggplot(sim.model2, bounds = c(model$bounds[1,]))
```

```
Initial covariance parameters: 1, 0.2
Checking gradients of objective function.
Derivative checker results: 0 error(s) detected.

eval_grad_f[ 1 ] =  1.077893e+00 ~  1.077893e+00   [6.146900e-09]
eval_grad_f[ 2 ] = -3.606823e+00 ~ -3.606824e+00   [5.225424e-08]

Estimated covariance parameters via MLE: 0.27, 0.3
```



Note that the constraints can be taken into account in the covariance parameter estimation by using `add.constr = TRUE`. However, in that case, experiments become more expensive and unstable since the Gaussian orthant probabilities in (7.1) have to be estimated via MC. As shown in Chapter 7, when the number of observations is large enough, the (unconstrained) ML provides a fair trade-off between speed and estimation accuracy. For a small number of observations, maximising the constrained ML could provide more accurate estimators (see numerical illustrations from Chapter 7).

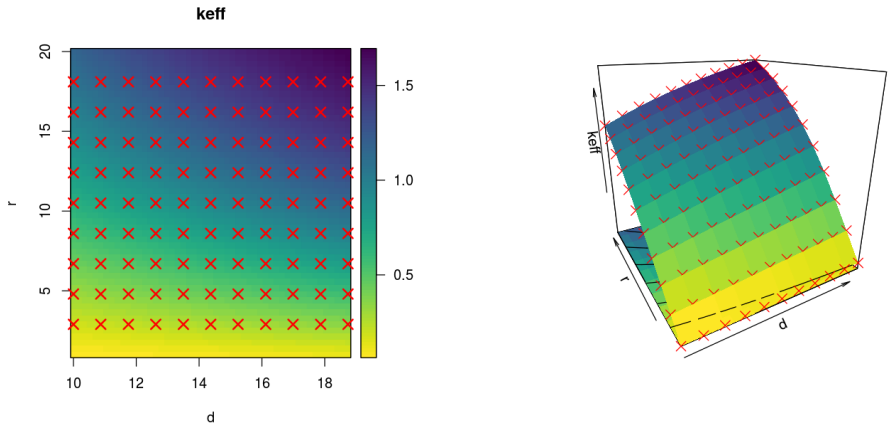
9.3 2D application: nuclear safety criticality

Finally, we aim at reproducing part of the results showed in Section 3.4 on the 2D nuclear application. We refer to that section for more details about the context of the problem.

In [10]:

```
library("lineqGPR")
library("DiceDesign")
library("viridis")
library("plot3D")
colormap <- rev(viridis(1e2))

#### Loading nuclear dataset from a local folder ####
data<-read.csv("lineqGPR_FullDemo_Docs/godiva.calculations.csv", sep=";")
data <- data[-1,] # removing the first line with data description
xgrid <- matrix(as.numeric(as.matrix(data[,c(1,2)])), ncol = 2)
ygrid <- data$mean_keff; x1 <- unique(xgrid[,1]); x2 <- unique(xgrid[,2])
```

```
## Plotting data ##
par(mfrow = c(1,2))
image2D(matrix(ygrid, nrow = length(x1)),  x1, x2, xlab = "d", ylab = "r",
        main = "keff", resfac = 5, col = colormap)
points(xgrid[,1], xgrid[,2], col= "red", pch = 4, cex = 1.5, lwd = 2)

p <- persp3D(x = x1, y = x2, z = matrix(ygrid, nrow = length(x1)),
             xlab = "d", ylab = "r", zlab = "keff", phi = 20, theta = -30,
             col = colormap, colkey = FALSE, image = TRUE, contour = TRUE)
points3D(x = xgrid[,1], y = xgrid[,2], z = ygrid, col = "red", pch = 4,
         cex = 1.5, add = TRUE)
```



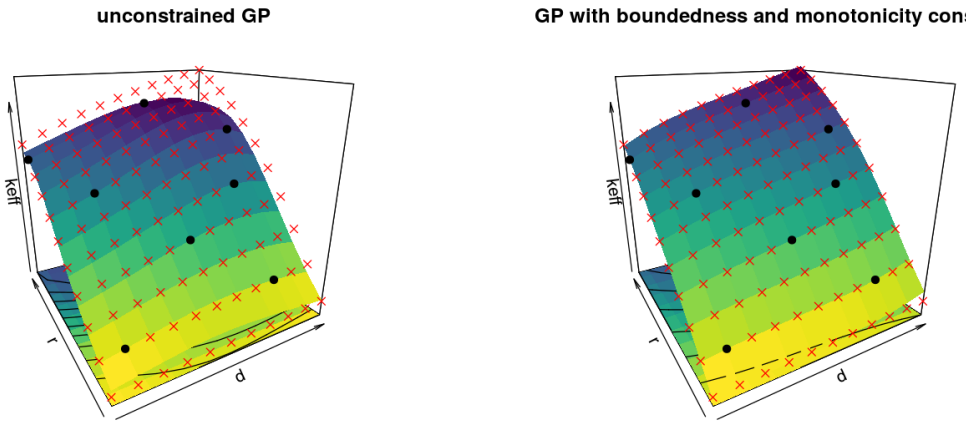
Here, we fit two types of GP models: (a) unconstrained constraints, and (b) accounting for both non-negativeness and monotonicity constraints. The covariance parameters of both models are estimated via ML.

In [11]:

```
## scaling input space to the unit square [0,1]^2
xgrid[,1] <- (xgrid[,1] - min(xgrid[,1]))/max(xgrid[,1] - min(xgrid[,1]))
xgrid[,2] <- (xgrid[,2] - min(xgrid[,2]))/max(xgrid[,2] - min(xgrid[,2]))
## fixing points for training and test the models
idx_train <- matrix(c(c(3,7), c(9,5), c(9,2), c(6,4), c(1,10),
                      c(2,2), c(10,7), c(7,10)), ncol = 2, byrow = TRUE)
idx_train <- length(x1)*(idx_train[,2]-1) + idx_train[,1]
xtrain <- xgrid[idx_train, ]; ytrain <- ygrid[idx_train]
xtest <- xgrid[-idx_train, ]; ytest <- ygrid[-idx_train]

#### Unconstrained GP ####
modelU <- create(class = "lineqGP", x = xtrain, y = ytrain,
                 constrType = "boundedness")
modelU$localParam$m <- 10 # number of knots per dimension
modelU$bounds <- c(lower = -10, upper = 10) # changing the bounds
modelU$kernParam$par <- c(sigma2 = 1^2, theta1 = 0.3, theta2 = 0.3)
```

```
    # simulating samples from the model
    sim.modelU <- simulate(modelU, nsim = 1e2, seed = 7, xtest = xgrid)

    #### GP with boundedness and monotonicity constraints ####
    modelC <- create(class = "lineqGP", x = xtrain, y = ytrain,
                     constrType = c("boundedness", "monotonicity"))
    modelC$localParam$m <- 10 # number of knots per dimension

    # modifying the bounds for first arg of "constrType" (boundedness)
    modelC$bounds[1,] <- c(lower = 0, upper = Inf)
    # changing the (default) covariance parameters
    modelC$kernParam$par <- c(sigma2 = 1^2, theta1 = 0.3, theta2 = 0.3)
    # simulating samples from the model
    sim.modelC <- simulate(modelC, nsim = 1e2, seed = 7, xtest = xgrid)

In [12]:
    #### Unconstrained GP ####
    message("Initial covariance parameters: ", modelU$kernParam$par[1],
            ", ", modelU$kernParam$par[2], ", ", modelU$kernParam$par[3])
    # estimating the covariance parameter via MLE for the unconstrained model
    modelU2 <- lineqGPOptim(modelU, eval_f = "logLik", add.constr = FALSE,
                            lb = c(0.1, 0.01, 0.01), ub = c(2, 0.5, 0.5))
    message("Estimated covariance parameters via MLE: ",
             modelU2$kernParam$par[1], ", ", modelU2$kernParam$par[2], ", ",
             modelU2$kernParam$par[3])
    # simulating samples from the model
    sim.modelU2 <- simulate(modelU2, nsim = 1e2, seed = 7, xtest = xgrid)

    # plotting the mean of these 100 simulations
    par(mfrow = c(1,2))
    p <- persp3D(x = x1, y = x2,
                 z = matrix(rowMeans(sim.modelU2$ysim), nrow = length(x1)),
                 xlab = "d", ylab = "r", zlab = "keff",
                 main = "unconstrained GP", zlim = c(0, max(ygrid)), phi = 20,
                 theta = -30, col = colormap, colkey = FALSE, image = TRUE,
                 contour = TRUE)
    points(trans3D(x = modelU2$x[,1], y = modelU2$x[,2], z = modelU2$y,
                   pmat = p), pch = 19)
    points(trans3D(x = xtest[,1], y = xtest[,2], z = ytest, pmat = p),
           col = "red", pch = 4)

    #### GP with boundedness and monotonicity constraints ####
    # estimating the covariance parameter via MLE for the constrained model
    modelC2 <- lineqGPOptim(modelC, eval_f = "logLik", add.constr = FALSE,
                            lb = c(0.1, 0.01, 0.01), ub = c(2, 0.5, 0.5))
```

```
# simulating samples from the model
sim.modelC2 <- simulate(modelC2, nsim = 1e2, seed = 7, xtest = xgrid)

# plotting the mean of these 100 simulations
p <- persp3D(x = x1, y = x2,
             z = matrix(rowMeans(sim.modelC2$ysim), nrow = length(x1)),
             xlab = "d", ylab = "r", zlab = "keff",
             main = "GP with boundedness and monotonicity constraints",
             zlim = c(0, max(ygrid)), phi = 20, theta = -30,
             col = colormap, colkey = FALSE, image = TRUE, contour = TRUE)
points(trans3D(x = modelC2$x[,1], y = modelC2$x[,2], z = modelC2$y,
               pmat = p), pch = 19)
points(trans3D(x = xtest[,1], y = xtest[,2], z = ytest, pmat = p),
       col = "red", pch = 4)

Initial covariance parameters: 1, 0.3, 0.3
Estimated covariance parameters via MLE: 0.61, 0.5, 0.5
```



Part VIII

Conclusions and Perspectives

Chapter 10

Coping with Dimensions: Basis Functions using Triangulations

Contents

10.1 Introduction

As discussed in Chapters 3 to 5, the tensor construction in (3.9) suffers of the curse of the dimensionality. Since the number of terms of that construction increases exponentially with the dimension d, numerical implementations in high dimensions becomes intractable (i.e. problems involving tens of input variables). As shown in Chapter 6, considering additivity in (constrained) GP developments allows us to easily scale the finite-dimensional representation in (3.1) to hundreds of input variables. Although the improvements of the additive GP model, and its fast implementation, it may fail in real-world applications dealing with non-additive behaviours (see, e.g., subsection 6.2.3). This drawback can be mitigated by considering block-additivity as discussed in Section 6.3. However, in that case, the tractability of the model will depend on the sizes of the partitions. According to experiments in Chapters 3 to 5, one could consider up to 5D blocks using the tensor structure in (3.9).

It is worth studying an alternative design of the knots that allows us to approximate functions in high dimensions without relying on tensor constructions. This paradigm is commonly dealt with in finite element methods (Brenner and Scott, 2007; Zienkiewicz et al., 2013). As an example, one may consider designs of the knots based on Delaunay

triangulations (De Loera et al., 2010; Delaunay, 1934; Okabe et al., 1992). In that case, since the number of terms in the triangulation does not increase exponentially with d, constructions based on triangular designs of the knots can be performed more efficiently for $d \geq 2$ compared to crude tensor constructions. Triangular structures have been further investigated in finite element methods in order to approximate functions involving tens of input variables (Barber et al., 1996; De Loera et al., 2010; Lee and Schachter, 1980).

In this chapter, the main contributions are twofold. First, we introduced a finite-dimensional approximation of GPs where functions are projected into piecewise affine triangles. Observations are mapped into the space of the Delaunay triangulation through a barycentric coordinate system (Munkres, 2018). Second, we establish the procedure to ensure inequality constraints everywhere (e.g. boundedness or monotonicity). As in Chapters 3 to 6, the posterior distribution is truncated Gaussian-distributed and can be approximated via MC/MCMC algorithms.

We demonstrate on 2D synthetic examples that the proposed framework together with the HMC sampler results in a promising approach that enjoys better scalability in high dimensions. Furthermore, since the distribution of the knots is not restricted to rectangular designs, one can consider freely placing knots in regions requiring a higher quality of resolution. Although numerical illustrations in this chapter are limited to 2D examples, one must note that it can be potentially extended to higher dimensions since the barycentric coordinate system and the Delaunay triangulation are general concepts that can be applied for $d > 2$.

10.2 Approximation of Gaussian processes using Delaunay triangulations

In chapters 3 to 5, a GP was approximated by a finite-dimensional GP consisting in its piecewise linear interpolation at a set of knots as in (3.1). The piecewise linearity is reached through the use of the hat basis functions in (3.2). Then main benefit of such representation is that ensuring inequality constraints over the knots is a sufficient condition to satisfy them everywhere in the input space. We now explore an alternative construction of basis functions that also preserves such piecewise linearity. Unlike the tensor construction in (3.9), the new representation allows us to consider triangular designs of the knots as shown in Figure 10.1. Since the number of terms in the triangular construction does not increase exponentially with d, this alternative approach could be performed more efficiently for $d \geq 2$ compared to the tensor construction in (3.9).

10.2.1 Finite-dimensional representation in 2D

Let $\{Y(\boldsymbol{x}); \boldsymbol{x} \in \mathcal{D}\}$ be a centred GP with covariance function k and compact input space $\mathcal{D} = [0,1]^2$. Consider a design of knots $(t_1^1, t_2^1), \cdots, (t_1^m, t_2^m) \in \mathbb{R}^2$. Then, define a finite-dimensional approximation GP, denoted by Y_m, as the piecewise linear interpolation of

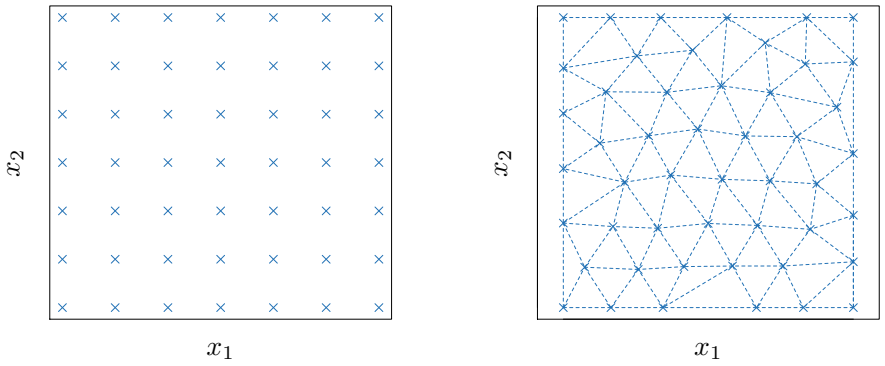


Figure 10.1: 2D visualisations of a tensor (left) and a triangular (right) design of knots.

Y at knots $(t_1^1, t_2^1), \cdots, (t_1^m, t_2^m)$:

$$Y_m(x_1, x_2) = \sum_{j=1}^{m} \xi_j \beta_j(x_1, x_2), \tag{10.1}$$

where $\xi_j = Y(t_1^j, t_2^j)$, and $\beta_1, \cdots, \beta_m$ are basis functions mapping Y_m into the space of the piecewise affine triangles constituted by a Delaunay triangulation (see Figure 10.2). To do so, we need to write Y_m in terms of its barycentric coordinates (De Loera et al., 2010; Munkres, 2018).

Definition 10.1 (Barycentric coordinate system.) *Let $\boldsymbol{x} = (x_1, \cdots, x_d) \in \mathbb{R}^d$. Consider $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_\nu$ the vertexes of a simplex in an affine space $\mathbb{R}^\nu$. Assume the condition $d = \nu - 1$. Then, the barycentric coordinates $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_\nu]^\top$ of the point $\boldsymbol{x}$, subject to $\sum_{j=1}^{\nu} \beta_j = 1$, are unique and can be obtained by solving the system of the $(d+1)$-coupled equalities:*

$$\begin{bmatrix} x_1 \\ \vdots \\ x_d \\ 1 \end{bmatrix} = \begin{bmatrix} v_1^1 & \cdots & v_1^\nu \\ \vdots & \ddots & \vdots \\ v_d^1 & \cdots & v_d^\nu \\ 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{\nu-1} \\ \beta_\nu \end{bmatrix}, \tag{10.2}$$

where v_i^j is the i-th component of the vertex $\boldsymbol{v}_i$. Furthermore, it is said that $\boldsymbol{x}$ is in the simplex formed by the vertexes $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_\nu$ if and only if all the components of $\boldsymbol{\beta}$ are strictly non-negatives, i.e. $0 \leq \beta_j \leq 1$ for $j = 1, \cdots, \nu$.

2D Illustration of the barycentric coordinate system. Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 \in [0,1]^2$ be three arbitrary points placed as in Figure 10.3. Consider $\boldsymbol{v}_1 = (0,0), \boldsymbol{v}_2 = (1,0), \boldsymbol{v}_3 = (1,1)$ te vertexes of a 2-simplex. Thus, the barycentric coordinates $\boldsymbol{\beta}_j = (\beta_1^j, \beta_2^j, \beta_3^j)$ of the j-th point $\boldsymbol{x}_j$, for $j = 1, 2, 3$, can be obtained by solving the linear system in

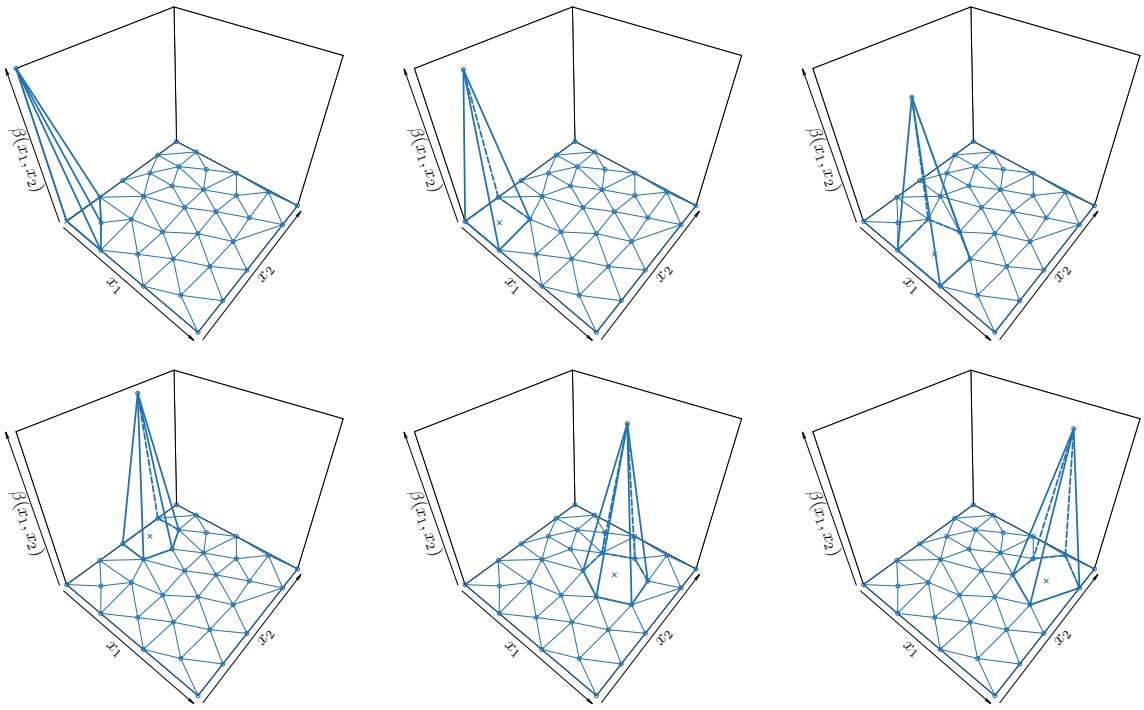


Figure 10.2: Examples of basis functions mapping to the space of the piecewise affine triangles constituted by a Delaunay triangulation.

(10.2). For the first point $\boldsymbol{x}_1 = (1, 0)$, we have the vector of barycentric coordinates $\boldsymbol{\beta}_1 = (0, 1, 0)$. This implies that there is only one predominant vertex and it corresponds to $\boldsymbol{v}_2$. Furthermore, due to the components of $\boldsymbol{\beta}_1$ being positive, one can conclude that $\boldsymbol{x}_1$ is inside the simplex. For the second point $\boldsymbol{x}_2 = (\frac{2}{3}, \frac{1}{3})$, we have that $\boldsymbol{\beta}_2 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Note that the barycentric components are equals when a point $\boldsymbol{x}$ corresponds to the centre of mass of a simplex. Finally, for the third point $\boldsymbol{x}_3 = (0, 1)$, the barycentric coordinates are $\boldsymbol{\beta}_3 = (1, -1, 1)$. Due to the second barycentric component of $\boldsymbol{\beta}_3$ is negative, one can confirm that $\boldsymbol{x}_3$ does not belong to the simplex.

2D Illustration of the construction of the barycentric basis functions. As shown in Figure 10.4, consider a DoE $\boldsymbol{x}_1 = (0.2, 0.3), \boldsymbol{x}_2 = (0.5, 0.8), \boldsymbol{x}_3 = (0.6, 0.3)$, and a design of knots $\boldsymbol{t}_1 = (0, 0), \boldsymbol{t}_2 = (0, 1), \boldsymbol{t}_3 = (0.6, 0.3), \boldsymbol{t}_4 = (1, 0), \boldsymbol{t}_5 = (1, 1)$. Using the barycentric decomposition in (10.2), then the representation in (10.1), evaluated at $\boldsymbol{x}_i$, for $i = 1, 2, 3$, follows the linear system of equalities:

$$\begin{bmatrix} Y_m(\boldsymbol{x}_1) \\ Y_m(\boldsymbol{x}_2) \\ Y_m(\boldsymbol{x}_3) \end{bmatrix} = \begin{bmatrix} \frac{7}{15} & \frac{1}{5} & \frac{1}{3} & 0 & 0 \\ 0 & \frac{27}{70} & \frac{2}{7} & 0 & \frac{23}{70} \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \end{bmatrix}.$$

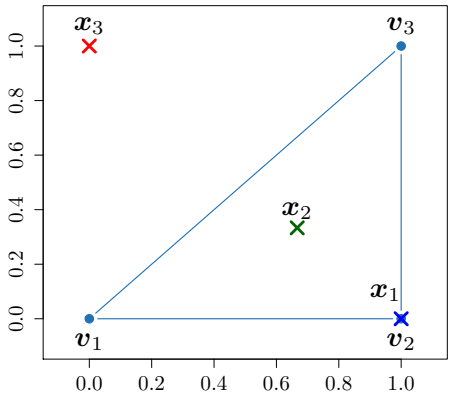


Figure 10.3: 2D Illustration of the barycentric coordinate system of three arbitrary points. Black solid lines represent the 2-simplex constituted by the vertexes $\boldsymbol{v}_1 = (0,0), \boldsymbol{v}_2 = (1,0)$ and $\boldsymbol{v}_3 = (1,1)$ (black dots). Multicolour crosses represent the points $\boldsymbol{x}_1 = (1,0)$ (blue), $\boldsymbol{x}_2 = (2/3, 1/3)$ (green), and $\boldsymbol{x}_3 = (0,1)$ (red).

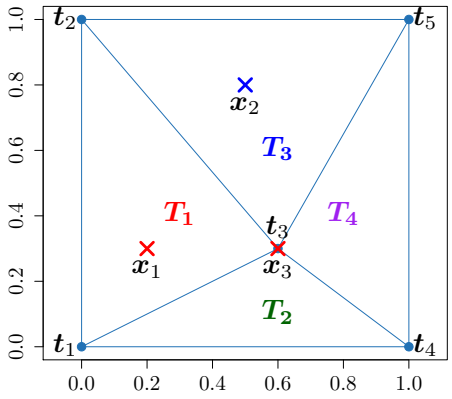


Figure 10.4: 2D Illustration of the Delaunay triangulation. Black solid lines represent the Delaunay triangulation of the knots $\boldsymbol{t}_j = (t_1^j, t_2^j)$ for $j = 1, \cdots, 5$ (black dots). The DoE $\boldsymbol{x}_1 = (0.2, 0.3), \boldsymbol{x}_2 = (0.5, 0.8), \boldsymbol{x}_3 = (0.6, 0.3)$ are shown by crosses and are associated to the triangles where they belong by colours.

Note that the barycentric coordinates of $Y_m(\boldsymbol{x}_i)$, for $i = 1, 2$, are non-zeros only over the knots forming the triangle that contains the point $\boldsymbol{x}_i$. For example, from Figure 10.4, one can observe that $\boldsymbol{x}_1$ belongs to the triangle $\boldsymbol{T}_1$ constituted by the knots at $\boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3$. Thus, $Y_m(\boldsymbol{x}_1)$ depends only on the values at the knots $\xi_1 = Y(\boldsymbol{t}_1), \xi_2 = Y(\boldsymbol{t}_2), \xi_3 = Y(\boldsymbol{t}_3)$. Note also that, since $\boldsymbol{x}_3 = \boldsymbol{t}_3$, then the values of $Y_m(\boldsymbol{x}_3)$ has to be exactly equal to $Y(\boldsymbol{t}_3)$.

10.2.2 Conditioning to interpolation and inequality constraints

We now consider the finite-dimensional representation of GPs as in (10.1), given the interpolation and inequality constraints:

$$Y_m(x_1, x_2) = \sum_{j=1}^{m} \xi_j \beta_j(x_1, x_2), \text{ s.t. } \begin{cases} Y_m(x_1^i, x_2^i) = y_i & \text{(interpolation conditions)}, \\ \boldsymbol{\xi} \in \mathcal{C} & \text{(inequality conditions)}, \end{cases} \tag{10.3}$$

where $(x_1^i, x_2^i) \in \mathcal{D}$ and $y_i \in \mathbb{R}$ for $i = 1, \cdots, n$. As discussed in Chapters 4 and 6, a noise effect can be included assuming $Y_m(x_1^i, x_2^i) + \varepsilon_i = y_i$ with Gaussian noise $\varepsilon_i \sim \mathcal{N}(0, \tau^2)$. Given a DoE $(x_1^1, x_2^1), \cdots, (x_1^n, x_2^n)$, we have matricially:

$$\mathbf{Y}_m = \left[Y_m(x_1^1, x_2^1), \quad \cdots, \quad Y_m(x_1^n, x_2^n)\right]^\top = \boldsymbol{\beta}\boldsymbol{\xi},$$

where $\boldsymbol{\beta}$ is the $n \times m$ matrix defined by $\boldsymbol{\beta}_{i,j} = \beta_j(x_1^i, x_2^i)$.

As in subsection 3.3.1, we aim at imposing linear inequality constraints defined in (3.10). Observe that the vector $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_m]^\top$ is a centred Gaussian vector with covariance matrix $\boldsymbol{\Gamma} = (k(t_1^i, t_2^i; t_1^j, t_2^j))_{1 \leq i,j \leq m}$. Then, the distribution of $\boldsymbol{\xi}$ given both interpolation and inequality conditions is truncated multinormal:

$$\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}) \quad \text{s.t.} \quad \begin{cases} \boldsymbol{\beta}\boldsymbol{\xi} = \boldsymbol{y} & \text{(interpolation conditions)}, \\ \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u} & \text{(inequality conditions)}, \end{cases} \tag{10.4}$$

where the matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{q \times m}$ encodes the linear operations, the vectors $\boldsymbol{l}$ and $\boldsymbol{u}$ represent the lower and upper bounds. As in subsection 3.3.1, one can show here that the posterior of (10.4) is truncated Gaussian-distributed: $\boldsymbol{\Lambda}\boldsymbol{\xi} | \{\boldsymbol{\beta}\boldsymbol{\xi} = \boldsymbol{y}, \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u}\} \sim \mathcal{TN}\left(\boldsymbol{\Lambda}\boldsymbol{\mu},\ \boldsymbol{\Lambda}\boldsymbol{\Sigma}\boldsymbol{\Lambda}^\top,\ \boldsymbol{l},\ \boldsymbol{u}\right)$ with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ as in (3.12) but replacing the hat basis functions $\boldsymbol{\Phi}$ by the new basis $\boldsymbol{\beta}$.

Finally, Algorithm 1 can be applied. Note that the knots and their triangulation have to be provided first in order to use Algorithm 1. Those knots can be strategically placed in the input domain according to diverse criteria. In our case, it is worth preferentially placing them in regions requiring a higher quality of resolution. Thus, as discussed in Chapter 5, an optimal criterion for free-knot and/or knot insertion algorithms is crucial. For instance, we consider minimax nearest-neighbour designs in order to cover all the input domain (see, e.g., Pronzato, 2017; Santner et al., 2003b). We refer to the theory behind finite-element methods for a further discussion on optimal space-filling algorithms (De Loera et al., 2010; Lee and Schachter, 1980; Pronzato and Müller, 2012).

We now discuss how to impose the constraints defined in (3.3). The case under boundedness constraints is straightforward since the piecewise linearity holds everywhere. We only need the condition of having knots at the corners of the input domain. The case under convexity conditions is more challenging and we let it as part of the future works.

Conditioning to monotonicity constraints. We consider the normal vectors generated by the affine simplexes constituted by the design of the knots. Let $\boldsymbol{\eta}_{i,j,k} = (\eta_1, \eta_2, \eta_3)_{i,j,k}$ be the normal vector associated to the simplex with vertexes $\boldsymbol{v}_i = (t_1^i, t_2^i, \xi_i)$, $\boldsymbol{v}_j = (t_1^j, t_2^j, \xi_j)$ and $\boldsymbol{v}_k = (t_1^k, t_2^k, \xi_k)$:

$$\boldsymbol{\eta}_{i,j,k} = \overline{\boldsymbol{v}}_{ij} \times \overline{\boldsymbol{v}}_{ik} = (\boldsymbol{v}_j - \boldsymbol{v}_i) \times (\boldsymbol{v}_k - \boldsymbol{v}_i),$$

that is equivalent to the linear system of equalities,

$$\begin{aligned} \eta_1^{i,j,k} &= \bar{t}_{jk}^2 \xi_i - \bar{t}_{ik}^2 \xi_j + \bar{t}_{ij}^2 \xi_k, \\ \eta_2^{i,j,k} &= -\bar{t}_{jk}^1 \xi_i + \bar{t}_{ik}^1 \xi_j - \bar{t}_{ij}^1 \xi_k, \\ \eta_3^{i,j,k} &= \bar{t}_{ij}^1 \bar{t}_{ik}^2 - \bar{t}_{ij}^2 \bar{t}_{ik}^1, \end{aligned} \tag{10.5}$$

with $\bar{t}_{ij}^r = (t_r^j - t_r^i)$ for $r = 1, 2$. For non-decreasing constraints, we need to ensure that the first two components of $\boldsymbol{\eta}_{i,j,k}$ satisfy the conditions,

$$\begin{aligned} -\operatorname{sign}(\eta_3^{i,j,k}) \eta_1^{i,j,k} &\geq 0, \\ -\operatorname{sign}(\eta_3^{i,j,k}) \eta_2^{i,j,k} &\geq 0. \end{aligned} \tag{10.6}$$

Observe that each inequality condition corresponds to non-decreasing assumptions along each dimension. Thus, monotonicity conditions can be assumed only w.r.t. one of the input coordinate by considering one of those conditions.

Using (10.5) and (10.6), we obtain the following linear system of inequalities for the affine triangle associated to the knots $(\boldsymbol{t}_i, \boldsymbol{t}_j, \boldsymbol{t}_k)$:

$$\begin{bmatrix} \lambda^2_{jk} & -\lambda^2_{ik} & \lambda^2_{ij} \\ -\lambda^1_{jk} & \lambda^1_{ik} & -\lambda^1_{ij} \end{bmatrix} \begin{bmatrix} \xi_i \\ \xi_j \\ \xi_k \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{10.7}$$

where $\lambda^r_{jk} = -\operatorname{sign}(\bar{t}^1_{ij}\bar{t}^2_{ik} - \bar{t}^2_{ij}\bar{t}^1_{ik})\bar{t}^r_{jk}$ for $r = 1, 2$. Let $N = 2m - m_c - 2$ be the total number of triangles with m the number of knots and m_c the number of knots at the boundaries. Thus, ensuring monotonicity conditions leads to a total number of inequalities $q = 2N = 4m - 2m_c - 4$ for $m \geq 4$. Finally, after defining all the sub-systems of inequalities as in (10.7) for the N triangles, one can establish the full linear system of inequalities $\boldsymbol{\Lambda}\boldsymbol{\xi} \geq \mathbf{0}$:

$$\mathcal{C}_1 = \left\{ \mathbf{c} \in \mathbb{R}^m;\ \forall\, k = 1, \ldots, 2N\ :\ \sum_{j=1}^{m} \lambda_{k,j} c_j \geq 0 \right\},$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{2N \times m}$ contains the $2N$ inequality conditions defined in (10.7). Since $2N > m$, and assuming that $\boldsymbol{\Lambda}$ has rank m, then $\boldsymbol{\Lambda}$ is injective (see, e.g., rank-nullity theorem in Meyer, 2000). Therefore, there exists a unique solution of $\boldsymbol{\Lambda}\boldsymbol{\xi} = \boldsymbol{\eta}$, and therefore, samples of $\boldsymbol{\xi}$ can be obtained using samples of $\boldsymbol{\Lambda}\boldsymbol{\xi}$.

10.2.3 Numerical illustrations

We revisit the 2D examples in Figure 3.5. We propose GP models under boundedness or monotonicity constraints with the same parametrisation suggested in subsection 3.3.3. For both examples, covariance parameters are fixed to $\boldsymbol{\theta} = (\sigma^2 = 1, \ell_1 = 0.2, \ell_2 = 0.2)$. We initialise the designs of knots using a random Latin hypercube DoE at 49 locations over $[0, 1]^2$, i.e. $m = 49$. We use a maximin nearest-neighbour space-filling technique in order to cover the input space. Then, we fit a Delaunay triangulation of the knots using the R package `deldir` (Turner, 2018). Results are shown in Figure 10.5 using the HMC sampler proposed in (Pakman and Paninski, 2014).

10.3 Conclusions

We have introduced a new finite-dimensional approximation of GPs using Delaunay triangulation of the knots. Here, the piecewise linear representation relies on the projection of functions into the space of affine simplexes through the barycentric coordinate system. In contrast to the tensor construction in (3.9), this alternative representation allows to freely place knots in the input space. Thus, one may allocate knots in regions requiring a higher quality of representation. Since the number of terms in the triangulation does not

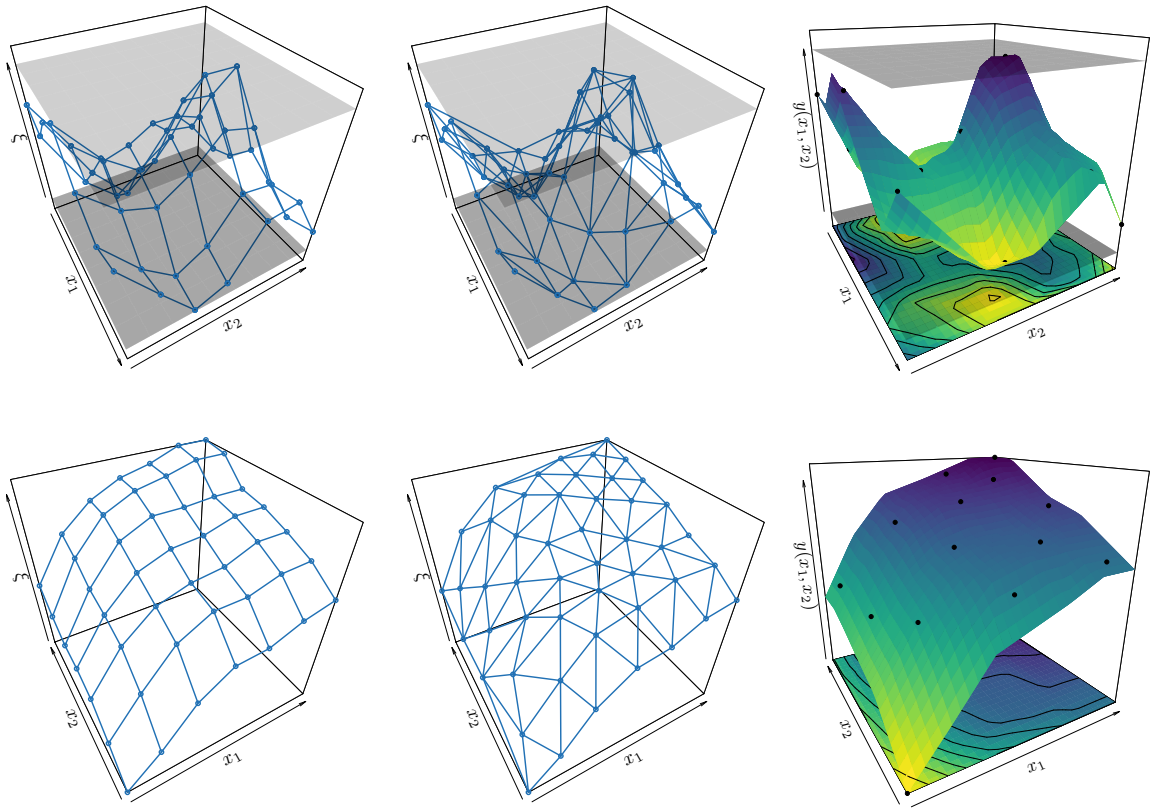


Figure 10.5: Examples of 2D GP models for interpolating the toy examples in Figure 3.5. Boundedness and monotonicity results are shown in the first and second row, respectively. Each row shows: the values at knots of the posterior mode provided by either the tensor construction (left) or triangular construction (centre), and the conditional mean function using the Delaunay design (right). Black dots represent the interpolation points. For boundedness constraints, (horizontal) grey surfaces denote the bounds.

increase exponentially with d, we believe that this new construction could be performed more efficiently for $d > 2$.

We demonstrated on 2D synthetic examples that the framework can satisfy boundedness and/or monotonicity constraints. We also believe that other types of inequality conditions can also be ensured (e.g. convexity constraints). Although numerical implementations were limited to two-dimensional input domains, the barycentric representation and the Delaunay triangulation are general concepts that can be used for $d > 2$.

As discussed throughout this chapter, the proposed framework still presents some limitations. First, the number of knots m was fixed aiming a trade-off between high resolution of representation and computational cost. As made for tensor-based constructions in Chapter 5, it is of interest having a sequential algorithm for knot insertion in highly variable regions. Second, we suggested a minimax nearest-neighbour space-

filling designs for the knots as a naive approach in order to cover the input domain. However, more dedicate constructions of triangular designs can be explored using the theory that has been developed for finite element methods. Finally, developments and numerical implementations for $d > 2$ are necessary in order to deal with a wider range of applications.

Chapter 11

Conclusions and Perspectives

11.1 Conclusions

This thesis was dedicated to the study of stochastic interpolation and regression models based on Gaussian processes (GPs) under inequality constraints (e.g. boundedness, monotonicity, convexity). The developments throughout this manuscript were in phase with three main directions: 1) to improve the applicability of GPs accounting for inequality constraints everywhere in the input space, 2) to make the constrained GP models scalable to higher dimensions and/or number of observations, and 3) to investigate parameter estimation under inequality constraints.

In Part III, we first gave answers to limitations of the framework proposed by Maatouk (2015). In Chapter 3, we extended the framework to deal with general sets of linear inequality constraints. This led to more versatile models that can be used for a broad range of real-world problems. As in (Maatouk, 2015), our framework resulted in a truncated Gaussian posterior distribution that can be sampled from with Monte Carlo (MC) and Markov Chain Monte Carlo (MCMC) algorithms. Thus, we explored efficient MC/MCMC samplers to make the model applicable on various numerical implementations. Finally, in Chapter 4, we also considered noisy observations. We concluded that adding an observation noise relaxed the interpolation conditions and resulted in a sample space with softer constraints. These improvements yielded a constrained framework that can be easily applicable on real-world problems up to 5D and to thousands of observations (e.g. nuclear safety criticality, coastal flooding)

In Part IV, we explored alternative constructions of finite-dimensional approximations of GPs to account for functions involving tens or hundreds of input variables. First, in Chapter 5, we investigated an alternative tensor construction that does not rely on equispaced designs of the knots. To do so, asymmetric hat basis functions were considered. Since the piecewise linearity holds in the new representation, all the properties of the framework in (Maatouk, 2015) were straightforwardly inherited. Furthermore, the non-equispaced construction allowed us to preferentially place knots only in regions requiring high quality of representation (typically in highly variable regions). For the knot location,

we introduced a sequential algorithm for the automatic knot insertion using an evolution criterion based on the maximisation of the integrated squared error of the MAP estimate. In Chapter 6, we considered additive (and block-additive) structures. In that case, we introduced a novel framework that enjoys the benefits of considering additive functions and that fulfills inequality constraints. Since constraints were assumed to be imposed on a predefined subset of input variables, developments proposed in Chapters 3 to 5 can be efficiently applied on (usually) low-dimensional subspaces. This led to constrained GP models that can be easily scaled in high dimensions involving hundreds of input variables.

Regarding inference under inequality constraints, Part V investigates a maximum likelihood (ML) estimator that accounts for inequality constraints. We showed that, loosely speaking, any consistency result for ML with unconstrained GPs is preserved when adding boundedness, monotonicity and convexity constraints. Furthermore, this consistency holds for both the unconditional and conditional likelihood functions. We also showed that both the unconstrained and constrained ML estimators (MLEs) are asymptotically Gaussian distributed if the GP satisfies either boundedness, monotonicity or convexity constraints. Their asymptotic distributions are identical to the unconditional asymptotic distribution of the MLE. This is not surprising since the model accurately "learns" the constraints when the number of observations n is large. In simulations, we observed that the constrained MLE is more accurate for small or moderate values of n.

In Part VI, we showed that developments of the previous chapters can be used for Cox processes, when the intensity function is modelled as a positive GP. The proposed approach can also ensure other types of inequality constraints (e.g. monotonicity, convexity), resulting in more versatile models that can be used for other classes of point processes (e.g. renewal processes).

Finally, Part VII presents the R package `lineqGPR`, which gathers the main methods that have been implemented throughout this manuscript. This is an important contribution for practical usage as well as a valuable tool for research. It is based on previous R packages (e.g. `DiceKriging`, `kergp`), but incorporating structures of classic GP libraries from other platforms (e.g. the `GPmat` toolbox from MATLAB, and the `GPy` library from Python). The current version on CRAN, `lineqGPR v.0.0.4`, contains implementations of Chapters 3, 4 and 7. Developments in Chapters 5, 6 and 10 have been added to a private beta version `lineqGPR v.0.1.0`.

11.2 Perspectives

As discussed throughout this manuscript, the GP developments proposed here can be improved in diverse aspects.

First, regarding the tensor representation in Chapters 3 to 5, the efficiency of the sequential algorithm based on the iMAP-SE criterion has been tested on various numerical illustrations. We believe that, as the number of knots goes to infinity, the rectangular design of knots provided by the algorithm will be dense in the input domain. From the fixed domain asymptotics' point of view, this assumption implies that the asymptotic

properties of the finite-dimensional approximation obtained by the symmetric hat basis functions hold for the asymmetric ones (i.e. convergence of the MAP estimate to the spline solution as shown in Bay et al., 2016). Moreover, alternative MAP-based criteria (e.g. integrated MAP absolute error) can be further investigated in future contributions.

Second, as discussed in Chapter 6, developments can be generalised to account for block-additivity. Since experiments there have been limited to additive structures without taking into account interactions between input variables, it is worth testing the proposed framework on block-additive applications. In that case, developments in Chapters 5 and 10 can be coupled.

Third, regarding the covariance parameter estimation, the constrained estimator suffers from certain drawbacks that limit its practical implementations (see Chapter 7 for a further discussion). First, since there is no closed form of the constrained likelihood, the evaluation and optimisation of constrained MLE have to be done numerically. Second, it requires the computation of Gaussian orthant probabilities in high dimensions. Existing methods for approximating those probabilities are time-consuming and present unstable results. Hence, further investigations are needed.

Fourth, as discussed in Chapter 8, one may also consider to couple the GP implementations along this manuscript to other types of GP-modulated processes satisfying certain inequality conditions (e.g. Gamma processes).

Part IX

Appendices

Chapter A

Maximum Likelihood Estimation under Inequality Constraints

In this appendix, we give the proof of the propositions and theorems from Chapter 7. For a self-containing reading, we write both the statements and proofs of the propositions and theorems. The proofs here have been obtained with the collaboration of F. Bachoc (IMT, Université Paul Sabatier, France) and A. Lagnoux (IMT, Université Toulouse Jean Jaurès, France).

A.1 Asymptotic consistency of ML estimators

Proposition A.1 *Let Y be a zero-mean GP on a bounded set $\mathbb{X} \subset \mathbb{R}^d$ with covariance function k satisfying Condition 7.1. Let $\boldsymbol{\Theta}$ be a compact set on $(0,\infty)^{d+1}$. Let $k_{\boldsymbol{\theta}}$ be the covariance function of $x \to \sigma Y(\alpha_1 x_1, \cdots, \alpha_d x_d)$ for $\boldsymbol{\theta} = (\sigma^2, \alpha_1, \cdots, \alpha_d) \in \boldsymbol{\Theta}$. Let $\boldsymbol{\theta}^* = (1, \cdots, 1)$. Remark that $k = k_{\boldsymbol{\theta}^*}$ and assume that $\boldsymbol{\theta}^* \in \boldsymbol{\Theta}$. Let $(\boldsymbol{x}_i)_{i\in\mathbb{N}}$ be a dense sequence in $\mathbb{X}$. Let $\boldsymbol{Y}_n = \left[Y(x_1), \cdots, Y(x_n) \right]^\top$. Let*

$$\mathcal{L}_n(\boldsymbol{\theta}) = -\frac{1}{2}\log(\det(\boldsymbol{R}_{\boldsymbol{\theta}})) - \frac{1}{2}\boldsymbol{Y}_n^\top \boldsymbol{R}_{\boldsymbol{\theta}}^{-1}\boldsymbol{Y}_n - \frac{n}{2}\log 2\pi,$$

with $\boldsymbol{R}_{\boldsymbol{\theta}} = (k_{\boldsymbol{\theta}}(x_i, x_j))_{1\leq i,j\leq n}$. Let $\widehat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathcal{L}_n(\boldsymbol{\theta})$. Assume that $\forall \varepsilon > 0$,

$$P(\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\| \geq \varepsilon) \xrightarrow[n\to\infty]{} 0.$$

Let $\kappa \in \{0,1,2\}$. Let $\mathcal{E}_\kappa$ be as in (7.3). Then, we have $P(Y \in \mathcal{E}_\kappa) > 0$ from Lemmas A.3 to A.5, and thus

$$P(\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\| \geq \varepsilon \mid Y \in \mathcal{E}_\kappa) \xrightarrow[n\to\infty]{} 0.$$

Proof. We have

$$P(\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\| \geq \ \varepsilon | \ Y \in \mathcal{E}_\kappa) = \frac{P(\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\| \geq \varepsilon, \ Y \in \mathcal{E}_\kappa)}{P(Y \in \mathcal{E}_\kappa)} \leq \frac{P(\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\| \geq \varepsilon)}{P(Y \in \mathcal{E}_\kappa)}.$$

Since $P(Y \in \mathcal{E}_\kappa) > 0$ is fixed, and $P(\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\| \geq \varepsilon) \xrightarrow[n\to\infty]{} 0$, the result follows. □

Proposition A.2 *We use the same notations and assumptions as in Proposition 7.1. Let $\kappa \in \{0, 1, 2\}$ be fixed. Let $P_{\boldsymbol{\theta}}$ be the distribution of Y with covariance function $k_{\boldsymbol{\theta}}$. Let*

$$\mathcal{L}_{\mathcal{C},n}(\boldsymbol{\theta}) = \mathcal{L}_n(\boldsymbol{\theta}) + \log P_{\boldsymbol{\theta}}(Y \in \mathcal{E}_\kappa | \, \mathbf{Y}_n) - \log P_{\boldsymbol{\theta}}(Y \in \mathcal{E}_\kappa).$$

Assume that $\forall \varepsilon > 0$ and $\forall M < \infty$,

$$P\left(\sup_{\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \geq \varepsilon} (\mathcal{L}_n(\boldsymbol{\theta}) - \mathcal{L}_n(\boldsymbol{\theta}^*)) \geq -M \right) \xrightarrow[n\to\infty]{} 0.$$

Then,

$$P\left(\sup_{\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \geq \varepsilon} (\mathcal{L}_{\mathcal{C},n}(\boldsymbol{\theta}) - \mathcal{L}_{\mathcal{C},n}(\boldsymbol{\theta}^*)) \geq -M \;\middle|\; Y \in \mathcal{E}_\kappa \right) \xrightarrow[n\to\infty]{} 0.$$

Consequently

$$\arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}_n(\boldsymbol{\theta}) \xrightarrow[n\to\infty]{P} \boldsymbol{\theta}^*, \qquad \text{and} \qquad \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}_{\mathcal{C},n}(\boldsymbol{\theta}) \xrightarrow[n\to\infty]{P|Y \in \mathcal{E}_\kappa} \boldsymbol{\theta}^*,$$

where $\xrightarrow[n\to\infty]{P}$ denotes the convergence in probability under the distribution of Y, and $\xrightarrow[n\to\infty]{P|Y \in \mathcal{E}_\kappa}$ denotes the convergence in probability under the distribution of Y given $Y \in \mathcal{E}_\kappa$.

Proof. For any fixed $\delta > 0$, since $\log(P_{\boldsymbol{\theta}}(Y \in \mathcal{E}_\kappa \mid \mathbf{Y}_n)) \leq 0$ for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, the quantity $\mathscr{P} = P\{\sup_{\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \geq \varepsilon} \log(P_{\boldsymbol{\theta}}(Y \in \mathcal{E}_\kappa \mid \mathbf{Y}_n)) - \log(P_{\boldsymbol{\theta}^*}(Y \in \mathcal{E}_\kappa \mid \mathbf{Y}_n)) \geq \delta \mid Y \in \mathcal{E}_\kappa\}$ satisfies

$$\begin{aligned}
\mathscr{P} \leq & P\left\{ -\log(P_{\boldsymbol{\theta}^*}(Y \in \mathcal{E}_\kappa \mid \mathbf{Y}_n)) \geq \delta \;\middle|\; Y \in \mathcal{E}_\kappa \right\} \\
= & P\left\{ P_{\boldsymbol{\theta}^*}(Y \in \mathcal{E}_\kappa \mid \mathbf{Y}_n) \leq \exp(-\delta) \;\middle|\; Y \in \mathcal{E}_\kappa \right\} \xrightarrow[n\to\infty]{} 0,
\end{aligned}$$

from Lemmas A.1 and A.2. Also, from Lemma A.6, there exists $\Delta > 0$ so that we have

$$\inf_{\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \geq \varepsilon} P_{\boldsymbol{\theta}}(Y \in \mathcal{E}_\kappa) \geq \Delta > 0,$$

so that

$$\sup_{\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \geq \varepsilon} -\log(P_{\boldsymbol{\theta}}(Y \in \mathcal{E}_\kappa)) + \log(P_{\boldsymbol{\theta}^*}(Y \in \mathcal{E}_\kappa)) \leq -\log(\Delta) < \infty.$$

Hence, the proposition follows. □

Lemma A.1 *Let $0 \leq \ell < u \leq \infty$. Let*

$$P_{n,\ell,u}(\mathbf{Y}_n) = P_{\boldsymbol{\theta}^*}(Y \in \mathcal{E}_0 | \; \mathbf{Y}_n).$$

Then, $\forall \varepsilon \geq 0$, we have

$$P(P_{n,\ell,u}(\mathbf{Y}_n) \leq 1 - \varepsilon \mid Y \in \mathcal{E}_0) \xrightarrow[n\to\infty]{} 0.$$

Proof. From Lemma A.3 we have $P(Y \in \mathcal{E}_0) > 0$. Hence, it is sufficient to show

$$P(P_{n,\ell,u}(\mathbf{Y}_n) \leq 1 - \varepsilon,\ Y \in \mathcal{E}_0) \xrightarrow[n\to\infty]{} 0.$$

The term $P_{n,\ell,u}(\mathbf{Y}_n)$, being a conditional expectation, is a martingale w.r.t. the σ-algebra generated by $Y(\boldsymbol{x}_1), \cdots, Y(\boldsymbol{x}_n)$. Furthermore, $0 \leq P_{n,\ell,u}(\mathbf{Y}_n) \leq 1$. Hence

$$P_{n,\ell,u}(\mathbf{Y}_n) \xrightarrow[n\to\infty]{a.s.} P(Y \in \mathcal{E}_0 \mid \mathcal{F}_\infty),$$

where $\mathcal{F}_\infty$ is the σ-algebra generated by $[Y(\boldsymbol{x}_i)]_{i\in\mathbb{N}}$ using Theorem 6.2.3 from (Kallenberg, 2002). Let μ_n and k_n be the mean and the covariance function (respectively) of Y given $\mathbf{Y}_n$. From proposition 2.8 in (Bect et al., 2016), the conditional distribution of Y given $\mathcal{F}_\infty$ is the distribution of a GP with mean function μ_∞ and covariance function k_∞. Furthermore, a.s., μ_n and k_n converge uniformly to μ_∞ and k_∞, respectively. Hence we can show simply that, because $(x_i)_{i\in\mathbb{N}}$ is dense in $\mathbb{X}$, we have a.s. $\mu_\infty = Y$ and k_∞ is the zero function. Hence a.s. if $Y \in \mathcal{E}_0$ holds, then

$$P(Y \in \mathcal{E}_0 \mid \mathcal{F}_\infty) = 1, \quad \text{so that} \quad P_{n,\ell,u}(\mathbf{Y}_n) \xrightarrow[n\to\infty]{} 1.$$

Hence by the dominated convergence theorem

$$P(P_{n,\ell,u}(\mathbf{Y}_n) \leq 1 - \varepsilon,\ Y \in \mathcal{E}_0) \xrightarrow[n\to\infty]{} 0.$$

□

Lemma A.2 *Let $\kappa = \{1, 2\}$. Let*

$$P_n(\,\mathbf{Y}_n) = P_{\boldsymbol{\theta}^*}(Y \in \mathcal{E}_\kappa |\ \mathbf{Y}_n).$$

Then, $\forall \varepsilon > 0$, we have

$$P(P_n(\,\mathbf{Y}_n) \leq 1 - \varepsilon \mid Y \in \mathcal{E}_\kappa) \xrightarrow[n\to\infty]{} 0.$$

Proof. The proof is the same as that of Lemma A.1. In particular, we remark that $\mathbb{1}_{Y\in\mathcal{E}_\kappa}$ is a measurable random variable, as Y has C^κ trajectories. □

Lemma A.3 *Let $\kappa = 0$. Assume that Condition 7.1 is satisfied. Then*

$$P(Y \in \mathcal{E}_0) > 0, \quad \textit{for} \quad -\infty \leq \ell < u \leq \infty.$$

Proof. We first prove that for any $\delta > 0$

$$P(\forall \boldsymbol{x} \in \mathbb{X} :\ |Y(\boldsymbol{x})| \leq \delta) > 0.$$

This result is true and appears implicitly in the literature about small ball estimates for GP (Li and Linde, 1999). We nevertheless provide a proof of it for self-consistency. Let

$(\boldsymbol{v}_i)_{i\in\mathbb{N}}$ be a dense sequence in $\mathbb{X}$. Let $\mathbf{Y}_v = \left[Y(\boldsymbol{v}_1), \cdots, Y(\boldsymbol{v}_n) \right]^\top$. Let μ_n and k_n be the mean and the covariance function of Y given $\mathbf{Y}_v$. Then we let

$$d^2_{k_n}(\boldsymbol{x}, \boldsymbol{x}') = \operatorname{var}\left\{ (Y(\boldsymbol{x}) - Y(\boldsymbol{x}')) | \mathcal{F}_n \right\},$$

where $\mathcal{F}_n = \sigma(Y(\boldsymbol{v}_1), \cdots, Y(\boldsymbol{v}_n))$. Note that, for a Gaussian vector, the conditional variance is deterministic, i.e. $\operatorname{var}\left\{ (Y(\boldsymbol{x}) - Y(\boldsymbol{x}')) | \mathcal{F}_n \right\} = \mathbb{E}\left\{ \operatorname{var}\left\{ (Y(\boldsymbol{x}) - Y(\boldsymbol{x}')) | \mathcal{F}_n \right\} \right\}$. Thus

$$d^2_{k_n}(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}\left\{ \operatorname{var}\left\{ (Y(\boldsymbol{x}) - Y(\boldsymbol{x}')) | \mathcal{F}_n \right\} \right\} \leq \operatorname{var}\left\{ (Y(\boldsymbol{x}) - Y(\boldsymbol{x}')) \right\} = d^2_k(\boldsymbol{x}, \boldsymbol{x}'),$$

from the law of total variance. Hence $N(\mathbb{X}, d_{k_n}, \rho) \leq N(\mathbb{X}, d_k, \rho)\ \forall \rho$. Also, from Theorem 2.10 in (Azaïs and Wschebor, 2009) (together with a union bound and using that $\max_{\boldsymbol{x}\in\mathbb{X}} Y(\boldsymbol{x})$ and $\max_{\boldsymbol{x}\in\mathbb{X}} [-Y(\boldsymbol{x})]$ have the same law) we have, with C a universal constant,

$$\begin{aligned}
\mathbb{E}\left\{ \max_{\boldsymbol{x}\in\mathbb{X}} |Y(\boldsymbol{x}) - \mu_n(\boldsymbol{x})| \right\} &\leq C \int_0^\infty \sqrt{\log(N(\mathbb{X}, d_{k_n}, \rho))}\, d\rho \\
&= C \int_0^{2\sqrt{\sup_{\boldsymbol{x}\in\mathbb{X}} k_n(\boldsymbol{x},\boldsymbol{x})}} \sqrt{\log(N(\mathbb{X}, d_{k_n}, \rho))}\, d\rho \\
&\leq C \int_0^{2\sqrt{\sup_{\boldsymbol{x}\in\mathbb{X}} k_n(\boldsymbol{x},\boldsymbol{x})}} \sqrt{\log(N(\mathbb{X}, d_k, \rho))}\, d\rho.
\end{aligned}$$

This last integral goes to 0 as $n \to \infty$ because $\sup_{\boldsymbol{x}\in\mathbb{X}} k_n(\boldsymbol{x}, \boldsymbol{x}) \to 0$ (see the proof of Lemma A.1), and because of Condition 7.1. Hence $\max_{\boldsymbol{x}\in\mathbb{X}} |Y(\boldsymbol{x}) - \mu_n(\boldsymbol{x})|$ goes to 0 in probability. Furthermore, $\mathscr{P} = P(\forall \boldsymbol{x} \in \mathbb{X},\ -\delta \leq Y(\boldsymbol{x}) \leq \delta)$ satisfies

$$\begin{aligned}
\mathscr{P} &\geq P\left(\forall \boldsymbol{x} \in \mathbb{X},\ -\frac{\delta}{2} \leq \mu_n(\boldsymbol{x}) \leq \frac{\delta}{2},\ -\frac{\delta}{2} \leq Y(\boldsymbol{x}) - \mu_n(\boldsymbol{x}) \leq \frac{\delta}{2} \right) \\
&= P\left(\forall \boldsymbol{x} \in \mathbb{X},\ -\frac{\delta}{2} \leq \mu_n(\boldsymbol{x}) \leq \frac{\delta}{2} \right) P\left(\forall \boldsymbol{x} \in \mathbb{X},\ -\frac{\delta}{2} \leq Y(\boldsymbol{x}) - \mu_n(\boldsymbol{x}) \leq \frac{\delta}{2} \right),
\end{aligned}$$

since the distribution of $Y - \mu_n$ does not depend on $\mathbf{Y}_v$. We now fix $n \in \mathbb{N}$ for which the second probability is non-zero (the existence is guaranteed from above). Then, the first probability is non-zero by continuity since, when $\mathbf{Y}_v = \mathbf{0}$, then μ_n is the zero function. Hence we have

$$P(\forall \boldsymbol{x} \in \mathbb{X} :\ |Y(\boldsymbol{x})| \leq \delta) > 0.$$

Let f be a C^∞ function on $\mathbb{R}^d$, square integrable, satisfying

$$\forall \boldsymbol{x} \in \mathbb{X}, \quad \ell + \delta \leq f(\boldsymbol{x}) \leq u - \delta,$$

for $\delta > 0$. (f exists for $\delta > 0$ small enough, and can be taken for instance as $f(\boldsymbol{x}) = \exp\{-\tau \|\boldsymbol{x} - \boldsymbol{x}_0\|^2\} \left[\frac{u+\ell}{2}\right]$ with $\tau > 0$ small enough, and for any $\boldsymbol{x}_0 \in \mathbb{X}$). Let Z be a

GP with covariance function k and mean function f. Then, from what we have shown before, we have

$$P(\forall \boldsymbol{x} \in \mathbb{X} : \ |Z(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \delta) > 0,$$

so that

$$P(\forall \boldsymbol{x} \in \mathbb{X} : \ \ell \leq Z(\boldsymbol{x}) \leq u) > 0.$$

From (Yadrenko, 1983) (p.138), as discussed by Stein (1999) (p.121), the Gaussian measures of Y and Z are equivalent. Thus

$$P(Y \in \mathcal{E}_0) = P(\forall \boldsymbol{x} \in \mathbb{X} : \ \ell \leq Y(\boldsymbol{x}) \leq u) > 0.$$

□

Lemma A.4 *Let $\kappa = 1$. Assume that Condition 7.1 is satisfied. Then*

$$P\left(Y \in \mathcal{E}_1\right) > 0.$$

Proof. We first prove that for any $\delta > 0$

$$P\left(\forall i = 1, \cdots, d, \ \forall \boldsymbol{x} \in \mathbb{X} : \ \left|\frac{\partial}{\partial x_i} Y(\boldsymbol{x})\right| \leq \delta\right) > 0.$$

We let $(\boldsymbol{v}_i)_{i \in \mathbb{N}}$ and $\mathbf{Y}_v$ be defined as in the proof of Lemma A.3. Then, as in this proof we can show that for $i = 1, \cdots, d$

$$\max_{\boldsymbol{x} \in \mathbb{X}} \left| \frac{\partial}{\partial x_i} Y(\boldsymbol{x}) - \mathbb{E}\left\{ \frac{\partial}{\partial x_i} Y(\boldsymbol{x}) \middle| \mathbf{Y}_v \right\} \right| \xrightarrow[n \to \infty]{P} 0.$$

Furthermore, $\mathscr{P} = P\left(\forall i = 1, \cdots, d, \ \forall \boldsymbol{x} \in \mathbb{X} : \ \left|\frac{\partial}{\partial x_i} Y(\boldsymbol{x})\right| \leq \delta\right)$ satisfies

$$\begin{aligned}
\mathscr{P} \geq P\Bigg(&\forall i = 1, \cdots, d, \ \forall \boldsymbol{x} \in \mathbb{X}, \ -\frac{\delta}{2} \leq \mathbb{E}\left\{ \frac{\partial}{\partial x_i} Y(\boldsymbol{x}) \middle| \mathbf{Y}_v \right\} \leq \frac{\delta}{2}, \\
&\forall i = 1, \cdots, d, \ \forall \boldsymbol{x} \in \mathbb{X}, \ -\frac{\delta}{2} \leq \frac{\partial}{\partial x_i} Y(\boldsymbol{x}) - \mathbb{E}\left\{ \frac{\partial}{\partial x_i} Y(\boldsymbol{x}) \middle| \mathbf{Y}_v \right\} \leq \frac{\delta}{2} \Bigg) \\
= P\Bigg(&\forall i = 1, \cdots, d, \ \forall \boldsymbol{x} \in \mathbb{X}, \ -\frac{\delta}{2} \leq \mathbb{E}\left\{ \frac{\partial}{\partial x_i} Y(\boldsymbol{x}) \middle| \mathbf{Y}_v \right\} \leq \frac{\delta}{2} \Bigg) \\
\times P\Bigg(&\forall i = 1, \cdots, d, \ \forall \boldsymbol{x} \in \mathbb{X}, \ -\frac{\delta}{2} \leq \frac{\partial}{\partial x_i} Y(\boldsymbol{x}) - \mathbb{E}\left\{ \frac{\partial}{\partial x_i} Y(\boldsymbol{x}) \middle| \mathbf{Y}_v \right\} \leq \frac{\delta}{2} \Bigg).
\end{aligned}$$

Notice that the last equality holds because the distribution of the process $\boldsymbol{x} \to \frac{\partial}{\partial x_i} Y(\boldsymbol{x}) - \mathbb{E}\left\{ \frac{\partial}{\partial x_i} Y(\boldsymbol{x}) | \mathbf{Y}_v \right\}$ does not depend on $\mathbf{Y}_v$. We now fix $n \in \mathbb{N}$ so that the second probability is non-zero (the existence is guaranteed from above). Then, the first probability is

non-zero by continuity since, when $\mathbf{Y}_v = \mathbf{0}$, then for $i = 1, \cdots, d$, $\mathbb{E}\left\{\frac{\partial}{\partial x_i} Y | \mathbf{Y}_v\right\}$ is the zero function. Hence, we have obtained

$$P\left(\forall i = 1, \cdots, d,\ \forall \boldsymbol{x} \in \mathbb{X} :\ \left|\frac{\partial}{\partial x_i} Y(\boldsymbol{x})\right| \leq \delta\right).$$

We now conclude the proof in the same way as for Lemma A.3. We consider the mean function

$$f(\boldsymbol{x}) = \left[\sum_{i=1}^{d} x_i\right] \exp\{-\tau \|\boldsymbol{x} - \boldsymbol{x}_0\|^2\},$$

with $\boldsymbol{x}_0 \in \mathbb{X}$ and $\tau > 0$. For τ small enough, f is C^∞, square integrable, and satisfies

$$\forall i = 1, \cdots, d, \quad \forall \boldsymbol{x} \in \mathbb{X}, \quad \frac{\partial}{\partial x_i} f(\boldsymbol{x}) \geq \frac{1}{2}.$$

Then, we conclude the proof as in the proof of Lemma A.3. □

Lemma A.5 *Let $\kappa = 2$. Assume that Condition 7.1 is satisfied. Then,*

$$P\left(Y \in \mathcal{E}_2\right) > 0.$$

Proof. We first prove that for any $\delta > 0$

$$P\left(\forall i, j = 1, \cdots, d,\ \forall \boldsymbol{x} \in \mathbb{X} :\ \left|\frac{\partial^2}{\partial x_i \partial x_j} Y(\boldsymbol{x})\right| \leq \delta\right) > 0.$$

This is done in a similar way as for showing $P\left(\forall i = 1, \cdots, d,\ \forall \boldsymbol{x} \in \mathbb{X} :\ \left|\frac{\partial}{\partial x_i} Y(\boldsymbol{x})\right| \leq \delta\right) > 0$ in the proof of Lemma A.4. We then conclude similarly as the rest of the proof this Lemma. In particular, we consider the mean function

$$f(\boldsymbol{x}) = \left[\sum_{i=1}^{d} x_i^2\right] \exp\{-\tau \|\boldsymbol{x} - \boldsymbol{x}_0\|^2\},$$

with $\boldsymbol{x}_0 \in \mathbb{X}$ and $\tau > 0$. Let $\lambda_{\inf}(M)$ be the smallest eigenvalue of a symmetric matrix M. Then, for τ small enough, f is C^∞, square integrable, and satisfies

$$\forall \boldsymbol{x} \in \mathbb{X}, \quad \lambda_{\inf}\left(\frac{\partial^2}{\partial \boldsymbol{x}^2} f(\boldsymbol{x})\right) \geq 1.$$

□

Lemma A.6 *Let $\kappa \in \{0, 1, 2\}$. Assume that Condition 7.1 holds. Let $Y_{\boldsymbol{\theta}}$ be the GP defined by*

$$Y_{\boldsymbol{\theta}}(t) = \sigma Y(\alpha_1 t_1, \cdots, \alpha_d t_d).$$

Let $P_{\boldsymbol{\theta}}^{\kappa} = P(Y_{\boldsymbol{\theta}} \in \mathcal{E}_\kappa)$ (see (7.3)). Then,

$$\inf_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} P_{\boldsymbol{\theta}}^{\kappa} > 0.$$

Proof. We do the proof for $\kappa = 2$. The proof for $\kappa = 0, 1$ is similar. Let $\varepsilon > 0$ and let

$$P^{\kappa}_{\boldsymbol{\theta},\varepsilon} = \mathbb{E}\left\{\mathbb{1}_{I(Y_{\boldsymbol{\theta}})\geq\varepsilon} + \frac{I(Y_{\boldsymbol{\theta}})}{\varepsilon}\mathbb{1}_{0\leq I(Y_{\boldsymbol{\theta}})\leq\varepsilon}\right\},$$

with $I(Y_{\boldsymbol{\theta}}) = \inf_{\boldsymbol{x}\in\mathbb{X}} \lambda_{\inf}\left(\frac{\partial^2}{\partial \boldsymbol{x}^2} Y_{\boldsymbol{\theta}}(\boldsymbol{x})\right)$. We have $P^{\kappa}_{\boldsymbol{\theta},\varepsilon} \leq P^{\kappa}_{\boldsymbol{\theta}}$ for any $\varepsilon > 0$. With the proof of Lemma A.5, we also obtain for $\varepsilon > 0$ small enough

$$\forall \boldsymbol{\theta} \in \boldsymbol{\Theta} \quad P^{\kappa}_{\boldsymbol{\theta},\varepsilon} > 0.$$

Hence, the proof is concluded, by compacity, if we show that $\boldsymbol{\theta} \to P^{\kappa}_{\boldsymbol{\theta},\varepsilon}$ is a continuous function on $\boldsymbol{\Theta}$. Let us show this. Let $\boldsymbol{\theta} = (\sigma_1^2, \alpha_1, \cdots, \alpha_d) \in (0,\infty)^{d+1}$ and $\boldsymbol{\theta}_n = (\sigma_n^2, \alpha_{n1}, \cdots, \alpha_{nd}) \to \boldsymbol{\theta}$. We have

$$\frac{\partial^2}{\partial x_i x_j} Y_{\boldsymbol{\theta}_n}(\boldsymbol{x}) = \sigma_n \left((\alpha_n)_i (\alpha_n)_j \frac{\partial^2}{\partial x_i x_j} Y(\alpha_{n1} x_1, \cdots, \alpha_{nd} x_d)\right).$$

Hence, because Y is C^2, we have a.s.

$$\sup_{\boldsymbol{x}\in\mathbb{X}} \left\| \frac{\partial^2}{\partial \boldsymbol{x}^2} Y_{\boldsymbol{\theta}_n}(\boldsymbol{x}) - \frac{\partial^2}{\partial \boldsymbol{x}^2} Y_{\boldsymbol{\theta}}(\boldsymbol{x}) \right\| \xrightarrow[n\to\infty]{} 0,$$

for any matrix norm $\|\cdot\|$. Hence also since Y is C^2, we can show, a.s.

$$\left(\inf_{\boldsymbol{x}\in\mathbb{X}} \lambda_{\inf}\left(\frac{\partial^2}{\partial \boldsymbol{x}^2} Y_{\boldsymbol{\theta}_n}(\boldsymbol{x})\right) - \inf_{\boldsymbol{x}\in\mathbb{X}} \lambda_{\inf}\left(\frac{\partial^2}{\partial \boldsymbol{x}^2} Y_{\boldsymbol{\theta}}(\boldsymbol{x})\right)\right) \xrightarrow[n\to\infty]{} 0.$$

Hence, we conclude by dominated convergence observing that $t \to (\mathbb{1}_{t\geq\varepsilon} + \frac{t}{\varepsilon}\mathbb{1}_{0\leq t\leq\varepsilon})$ is a continuous function on $\mathbb{R}$. □

A.2 Asymptotic normality of ML estimators

A.2.1 Notation

Here, $0 < c < +\infty$ stands for a generic constant that may differ from one line to another. It is convenient to have short expressions for terms that converge in probability to zero. Following (Van der Vaart, 1998), the notation $o_{\mathbb{P}}(1)$ (respectively $O_{\mathbb{P}}(1)$) stands for a sequence of random variables (r.v.'s) that converges to zero in probability (resp. is bounded in probability) as $n \to \infty$. More generally, for a sequence of r.v.'s R_n,

$$\begin{aligned} X_n = o_{\mathbb{P}}(R_n) \quad &\text{means} \quad X_n = Y_n R_n \quad \text{with} \quad Y_n \xrightarrow{P} 0, \\ X_n = O_{\mathbb{P}}(R_n) \quad &\text{means} \quad X_n = Y_n R_n \quad \text{with} \quad Y_n = O_{\mathbb{P}}(1). \end{aligned}$$

For deterministic sequences X_n and R_n, the stochastic notation reduce to the usual o and O. For a sequence of random vectors or variables $(X_n)_{n\in\mathbb{N}}$ on $\mathbb{R}^l$, that are functions of Y, and for a probability distribution μ on $\mathbb{R}^l$, we write

$$X_n \xrightarrow[n\to\infty]{\mathcal{L}|Y\in\mathcal{E}_\kappa} \mu$$

when, for any bounded continuous function $g : \mathbb{R}^l \to \mathbb{R}$, we have

$$\mathbb{E}\left[g(X_n)|\, Y \in \mathcal{E}_\kappa\right] \underset{n\to\infty}{\longrightarrow} \int_{\mathbb{R}^l} g(x)\mu(dx).$$

We also write $X_n = o_{\mathbb{P}|Y\in\mathcal{E}_\kappa}(1)$ when for all $\varepsilon > 0$ we have $\mathbb{P}(|X_n| \geqslant \varepsilon | Y \in \mathcal{E}_\kappa) \to 0$ as $n \to \infty$.

For any two functions $f(Y)$ and $g(Y)$, let $\mathbb{E}_\theta[f(Y)]$ (respectively $\mathbb{E}_\theta[f(Y)|g(Y)]$) be the expectation (resp. the conditional expectation) w.r.t. the measure $\mathbb{P}_\theta$ on Ω. We define similarly $\mathbb{P}_\theta(A(Y))$ and $\mathbb{P}_\theta(A(Y)|g(Y))$ when $A(Y)$ is an event w.r.t. Y. Let $\theta_0 \in \Theta$ be fixed. We consider θ_0 as the true unknown covariance parameter and we let $\mathbb{E}[\cdot]$, $\mathbb{E}[\cdot|\cdot]$, $\mathbb{P}(\cdot)$, and $\mathbb{P}(\cdot|\cdot)$ be shorthands for $\mathbb{E}_{\theta_0}[\cdot]$, $\mathbb{E}_{\theta_0}[\cdot|\cdot]$, $\mathbb{P}_{\theta_0}(\cdot)$, and $\mathbb{P}_{\theta_0}(\cdot|\cdot)$. When a quantity is said to converge, say, in probability or almost surely, it is also implicit that we consider the measure $\mathbb{P}_{\theta_0}$ on Ω.

For $a > 0$, let $f_a : (0,\infty) \to \mathbb{R}$ be defined by $f_a(t) = -\log(t) - a/t$. We will repeatedly use the fact that f_a has a unique global maximum at a and $f_a''(t) = 1/t^2 - 2a/t^3$.

Finally, let $\xi_* = \underset{x\in[0,1]^d}{inf}\ \xi(x)$, $\xi^* = \underset{x\in[0,1]^d}{sup}\ \xi(x)$, and $\xi^{**} = \underset{x\in[0,1]^d}{sup}\ |\xi(x)|$ for any stochastic process $\xi : [0,1]^d \to \mathbb{R}$.

A.2.2 Intermediate results

Lemma A.7 *Let $(X_n)_{n\in\mathbb{N}}$ be a sequence of r.v.'s and $(m_{k,n})_{n,k\in\mathbb{N},\, k\leqslant n}$ and $(M_{k,n})_{n,k\in\mathbb{N},\, k\leqslant n}$ be two triangular arrays of r.v.'s. We consider a random vector $(m, M)^\top$ such that $m \leqslant m_{k,n} \leqslant M_{k,n} \leqslant M$ for all $k \leqslant n$. We assume that $\mathbb{P}(m = \ell) = \mathbb{P}(M = u) = 0$ and $\mathbb{P}(\ell \leqslant m \leqslant M \leqslant u) > 0$ for some fixed ℓ and $u \in \mathbb{R}$. Moreover, we consider a sequence $(k_n)_{n\in\mathbb{N}}$ so that, $k_n \leqslant n$, $k_n \to_{n\to\infty} \infty$ and*

$$(m_{k_n,n}, M_{k_n,n})^\top \xrightarrow[n\to+\infty]{a.s.} (m, M)^\top. \tag{A.1}$$

Then for any $a \in \mathbb{R}$,

$$\lim_{n\to+\infty} \Big|\mathbb{P}(X_n \leqslant a | \ell \leqslant m_{k_n,n} \leqslant M_{k_n,n} \leqslant u) - \mathbb{P}(X_n \leqslant a | \ell \leqslant m \leqslant M \leqslant u)\Big| = 0. \tag{A.2}$$

Proof. For the sake of simplicity, we denote by $E_{k,n}$ (respectively E) the event $\{\ell \leqslant m_{k,n} \leqslant M_{k,n} \leqslant u\}$ (resp. $\{\ell \leqslant m \leqslant M \leqslant u\}$). Then

$$\begin{aligned}|\mathbb{P}(X_n \leqslant a | E_{k_n,n}) - \mathbb{P}(X_n \leqslant a | E)| \leqslant & \frac{|\mathbb{P}(X_n \leqslant a,\ E_{k_n,n}) - \mathbb{P}(X_n \leqslant a,\ E)|}{\mathbb{P}(E_{k_n,n})} \\ & + \left|\frac{1}{\mathbb{P}(E_{k_n,n})} - \frac{1}{\mathbb{P}(E)}\right| \mathbb{P}(X_n \leqslant a,\ E).\end{aligned} \tag{A.3}$$

(i) By (A.1), $\mathbb{P}(E_{k_n,n})$ goes to $\mathbb{P}(E) = \mathbb{P}(\ell \leqslant m \leqslant M \leqslant u) > 0$ as n goes to $+\infty$. Thus $1/\mathbb{P}(E_{k_n,n})$ is well-defined for large values of n and bounded as $n \to \infty$. Moreover, by trivial arguments of set theory, one gets

$$|\mathbb{P}(X_n \leqslant a,\ E_{k_n,n}) - \mathbb{P}(X_n \leqslant a,\ E)| \leqslant \mathbb{P}(E_{k_n,n} \Delta E) = \mathbb{P}(E_{k_n,n} \setminus E)$$

since $\mathbb{P}(E \setminus E_{k_n,n}) = 0$. Now let $\varepsilon > 0$. One has

$$\begin{aligned}
\mathbb{P}(E_{k_n,n} \setminus E) &= \mathbb{P}(\ell \leqslant m_{k_n,n} \leqslant M_{k_n,n} \leqslant u,\ (m, M) \notin [\ell, u]^2) \\
&\leqslant \mathbb{P}(\ell \leqslant m_{k_n,n} \leqslant M_{k_n,n} \leqslant u,\ m < \ell) + \mathbb{P}(\ell \leqslant m_{k_n,n} \leqslant M_{k_n,n} \leqslant u,\ M > u) \\
&\leqslant \mathbb{P}(\ell \leqslant m_{k_n,n},\ m < \ell) + \mathbb{P}(M_{k_n,n} \leqslant u,\ M > u).
\end{aligned}$$

One may decompose $\mathbb{P}(\ell \leqslant m_{k_n,n},\ m < \ell)$ into

$$\begin{aligned}
&\mathbb{P}(\ell + \varepsilon \leqslant m_{k_n,n},\ m < l) + \mathbb{P}(\ell \leqslant m_{k_n,n} \leqslant \ell + \varepsilon,\ m < l) \\
&\quad \leqslant \mathbb{P}(|m_{k_n,n} - m| > \varepsilon) + \mathbb{P}(\ell \leqslant m_{k_n,n} \leqslant \ell + \varepsilon).
\end{aligned}$$

The first term in the right hand-side goes to 0 as n goes to infinity. By Portemanteau's lemma and (A.1),

$$\limsup_{n \to +\infty} \mathbb{P}(\ell \leqslant m_{k_n,n} \leqslant \ell + \varepsilon) \leqslant \mathbb{P}(\ell \leqslant m \leqslant \ell + \varepsilon) \underset{\varepsilon \to 0}{\longrightarrow} 0.$$

We handle similarly the term $\mathbb{P}(M_{k_n,n} \leqslant u,\ M > u)$. Hence, in the r.h.s. of (A.3), the first term goes to 0 as $n \to \infty$.

(ii) Now we turn to the control of the second term in (A.3). Upper bounding $\mathbb{P}(X_n \leqslant a,\ E)$ by 1, it remains to control $\left| \frac{1}{\mathbb{P}(E_{k_n,n})} - \frac{1}{\mathbb{P}(E)} \right|$ which is immediate by the convergence in distribution of $(m_{k_n,n}, M_{k_n,n})^\top$ as n goes to infinity (implied by the a.s. convergence) and the fact that $\mathbb{P}(E) > 0$ and $\mathbb{P}(m = \ell) = \mathbb{P}(M = u) = 0$. The proof is now complete. □

Lemma A.8 *Consider three sequences of random functions $f_n, g_n, h_n : [x_{inf}, x_{sup}] \to \mathbb{R}$, with $0 < x_{inf} < x_{sup} < \infty$ fixed. Consider that for all $x \in [x_{inf}, x_{sup}]$, $f_n(x)$, $g_n(x)$, and $h_n(x)$ are functions of Y and x only. Let*

$$\widehat{x}_n \in \underset{x \in [x_{inf}, x_{sup}]}{arg\ max}\ f_n(x).$$

Assume the following properties.

(i) There exists $A > 0$, $B > 0$ and $\delta > 0$ such that

$$\underset{\substack{|x - \widehat{x}_n| \leqslant \delta \\ x \in [x_{inf}, x_{sup}]}}{sup} f_n(x) - f_n(\widehat{x}_n) \leqslant -An(x - \widehat{x}_n)^2 \tag{A.4}$$

and

$$\underset{\substack{|x - \widehat{x}_n| > \delta \\ x \in [x_{inf}, x_{sup}]}}{sup} f_n(x) - f_n(\widehat{x}_n) \leqslant -Bn, \tag{A.5}$$

with probability going to 1 as $n \to \infty$.

(ii) *There exists* $C > 0$ *such that for all* $x_1, x_2 \in [x_{inf}, x_{sup}]$

$$|g_n(x_1) - g_n(x_2)| \leqslant C|x_1 - x_2|, \tag{A.6}$$

with probability going to 1 *as* $n \to \infty$.

(iii) *One has, for* $\kappa = 0, 1, 2$,

$$\sup_{x_1,x_2\in[x_{inf},x_{sup}]} |h_n(x_1) - h_n(x_2)| = o_{\mathbb{P}|Y\in\mathcal{E}_\kappa}(1). \tag{A.7}$$

Then, with

$$\widehat{\widehat{x}}_n \in \underset{x\in[x_{inf},x_{sup}]}{arg\ max} \{f_n(x) + g_n(x) + h_n(x)\}$$

we have

$$\sqrt{n}|\widehat{\widehat{x}}_n - \widehat{x}_n| = o_{\mathbb{P}|Y\in\mathcal{E}_\kappa}(1). \tag{A.8}$$

Proof. Let $\varepsilon > 0$. First, we have, with probability (conditionally to $\{Y \in \mathcal{E}_\kappa\}$) going to 1 as $n \to \infty$, from (A.4), (A.6), and (A.7)

$$\sup_{\substack{|x-\widehat{x}_n|\geqslant\varepsilon/\sqrt{n}\\ |x-\widehat{x}_n|\leqslant 1/n^{1/4}}} (f_n(x) + g_n(x) + h_n(x) - f_n(\widehat{x}_n) - g_n(\widehat{x}_n) - h_n(\widehat{x}_n))$$
$$\leqslant -An\left(\frac{\varepsilon}{\sqrt{n}}\right)^2 + \frac{C}{n^{1/4}} + o_{\mathbb{P}Y\in\mathcal{E}_\kappa}(1) = -A\varepsilon^2 + o_{\mathbb{P}|Y\in\mathcal{E}_\kappa}(1).$$

Second, from (A.4), (A.6), and (A.7), we have, with probability (conditionally to $\{Y \in \mathcal{E}_\kappa\}$) going to 1 as $n \to \infty$,

$$\sup_{\substack{|x-\widehat{x}_n|\geqslant 1/n^{1/4}\\ |x-\widehat{x}_n|\leqslant\delta}} (f_n(x) + g_n(x) + h_n(x) - f_n(\widehat{x}_n) - g_n(\widehat{x}_n) - h_n(\widehat{x}_n))$$
$$\leqslant -An\left(\frac{1}{n^{1/4}}\right)^2 + C\delta + o_{\mathbb{P}|Y\in\mathcal{E}_\kappa}(1) \underset{n\to\infty}{\longrightarrow} -\infty.$$

Third, from (A.5)–(A.7), we have, with probability (conditionally to $\{Y \in \mathcal{E}_\kappa\}$) going to 1 as $n \to \infty$,

$$\sup_{|x-\widehat{x}_n|\geqslant\delta} (f_n(x) + g_n(x) + h_n(x) - f_n(\widehat{x}_n) - g_n(\widehat{x}_n) - h_n(\widehat{x}_n))$$
$$\leqslant -Bn + C(x_{sup} - x_{inf}) + o_{\mathbb{P}|Y\in\mathcal{E}_\kappa}(1) \underset{n\to\infty}{\longrightarrow} -\infty.$$

Finally, for all $\varepsilon > 0$ there exists $c > 0$ so that, with probability (conditionally to $\{Y \in \mathcal{E}_\kappa\}$) going to 1 as $n \to \infty$,

$$\sup_{|x-\widehat{x}_n|\geqslant\varepsilon/\sqrt{n}} (f_n(x) + g_n(x) + h_n(x) - f_n(\widehat{x}_n) - g_n(\widehat{x}_n) - h_n(\widehat{x}_n)) \leqslant -c.$$

Hence, we have, by definition of $\widehat{\widehat{x}}_n$

$$\sqrt{n}|\widehat{\widehat{x}}_n - \widehat{x}_n| = o_{\mathbb{P}|Y\in\mathcal{E}_\kappa}(1).$$

□

Lemma A.9 *Let $\{k_\theta; \theta \in \Theta\}$ be the set of covariance functions in subsection 2.2.3 where Θ is compact. Assume that k_θ satisfies Condition 7.4 in the case $\kappa = 0$, where C and α can be chosen independently of θ. Let $Z_{n,\theta}$ be a Gaussian process with mean function zero and covariance function $(x_1, x_2) \mapsto \mathrm{Cov}_\theta(Y(x_1), Y(x_2)|y)$. Then, we have*

$$\sup_{\theta\in\Theta} \mathbb{E}\left[\sup_{x\in[0,1]^d} |Z_{n,\theta}(x)|\right] \underset{n\to\infty}{\longrightarrow} 0.$$

Proof. This result is proved as an intermediate result in the proof of Lemma A.3 in (López-Lopera et al., 2018). There, the result was for fixed θ, but it can be made uniform over $\theta \in \Theta$ with no additional difficulties. □

A.2.3 Variance parameter estimation

Here, we display the proof of the theorems proposed in Section 7.4. For the sake of compaction, we write only the proofs for results in subsection 7.4.1 under boundedness constraints, i.e. we let $\kappa = 0$. We refer to (Bachoc et al., 2019) for additional proofs for mototonicity ($\kappa = 1$) and convexity ($\kappa = 2$) constraints, and for the microergodic parameter estimation for the isotropic Matérn model (see subsection 7.4.2).

Lemma A.10 *Assume that Condition 7.4 holds. Then for all $\kappa \in \{0, 1, 2\}$ and for any compact K in $(0, +\infty)$, we have*

$$\inf_{\sigma^2\in K} \mathbb{P}_{\sigma^2}(Y \in \mathcal{E}_\kappa) > 0.$$

Proof. It suffices to follow the same lines as in the proof of Lemma A.6. □

Theorem A.1 *For $\kappa = 1, 2$, we assume that Condition 7.2 holds. Under Condition 7.4, the MLE $\bar{\sigma}_n^2$ of σ_0^2 defined by (7.11) conditioned on $\{Y \in \mathcal{E}_\kappa\}$ is asymptotically Gaussian distributed. More precisely, for $\kappa = 0, 1, 2$,*

$$\sqrt{n}\left(\bar{\sigma}_n^2 - \sigma_0^2\right) \xrightarrow[n\to+\infty]{\mathcal{L}|Y\in\mathcal{E}_\kappa} \mathcal{N}(0, 2\sigma_0^4).$$

Proof.

(1) Let $m_{k,n} = \min_{i=1,\dots,k} y_i$, $M_{k,n} = \max_{i=1,\dots,k} y_i$, and $(m, M)^\top = (Y_*, Y^*)^\top$, where Y_* and Y^* have been defined in Appendix A.2.1. We clearly have $m \leqslant m_{k_n,n} \leqslant M_{k,n} \leqslant M$. Since $(x_i)_{i\in\mathbb{N}}$ is dense, for any sequence $(k_n)_{n\in\mathbb{N}}$ so that $k_n \to \infty$ as $n \to \infty$ and $k_n \leqslant n$, we

have $(m_{k_n,n}, M_{k_n,n})^\top \to (m, M)$ a.s. as $n \to \infty$ (up to reindexing $x_1, \ldots, x_n$).

2) Let $k \in \mathbb{N}$ be fixed. We have

$$\sqrt{n}\left(\bar{\sigma}_n^2 - \sigma_0^2\right) = \frac{1}{\sqrt{n}}\left(y^\top R_1^{-1} y - n\sigma_0^2\right).$$

Writing the Gaussian probability density function of y as the product of the conditional probability density functions of y_i given $y_1, \ldots, y_{i-1}$ leads to

$$\frac{1}{\sqrt{n}}\left(y^\top R_1^{-1} y - n\sigma_0^2\right) = \frac{\sigma_0^2}{\sqrt{n}} \sum_{i=1}^{n} \left(\frac{(y_i - \mathbb{E}[y_i|y_1, \ldots, y_{i-1}])^2}{\mathrm{Var}(y_i|y_1, \ldots, y_{i-1})} - 1 \right).$$

The terms in the sum above are independent. Indeed,

$$\mathrm{Cov}(y_l, y_i - \mathbb{E}[y_i|y_1, \ldots, y_{i-1}]) = 0, \quad \text{for any} \quad l \leqslant i-1$$

and the Gaussianity then leads to independence. Therefore,

$$\begin{aligned}\frac{1}{\sqrt{n}}\left(y^\top R_1^{-1} y - n\sigma_0^2\right) = &\frac{\sigma_0^2}{\sqrt{n}} \sum_{i=1}^{k} \left(\frac{(y_i - \mathbb{E}[y_i|y_1, \ldots, y_{i-1}])^2}{\mathrm{Var}(y_i|y_1, \ldots, y_{i-1})} - 1 \right) \\ &+ \frac{\sigma_0^2}{\sqrt{n}} \sum_{i=k+1}^{n} \left(\frac{(y_i - \mathbb{E}[y_i|y_1, \ldots, y_{i-1}])^2}{\mathrm{Var}(y_i|y_1, \ldots, y_{i-1})} - 1 \right).\end{aligned}$$

The first term is $o_{\mathbb{P}}(1)$ being the sum of k r.v.'s (whose variances are all equal to 2) divided by the square root of n. Because $\mathbb{P}_{\sigma^2}(\ell \leqslant \min\limits_{i=1,\ldots,k} y_i \leqslant \max\limits_{i=1,\ldots,k} y_i \leqslant u) > 0$, the first term is also $o_{\mathbb{P}}(1)$ conditionally to $\left\{\ell \leqslant \min\limits_{i=1,\ldots,k} y_i \leqslant \max\limits_{i=1,\ldots,k} y_i \leqslant u\right\}$. The second term is equal to $\sigma_0^2/\sqrt{n}$ times the sum of $n-k$ independent variables with zero mean and variance 2 and is also independent of $y_1, \ldots, y_k$. Hence, from the central limit theorem and Slutsky's lemma (Van der Vaart, 1998, Lemma 2.8), we obtain that

$$\frac{1}{\sqrt{n}}\left(y^\top R_1^{-1} y - n\sigma_0^2\right) \xrightarrow[n\to\infty]{\mathcal{L}|y\in\mathcal{E}_{0,k}} \mathcal{N}(0, 2\sigma_0^4),$$

where $\mathcal{E}_{0,k} := \left\{y : \ \ell \leqslant \min\limits_{i=1,\ldots,k} y_i \leqslant \max\limits_{i=1,\ldots,k} y_i \leqslant u\right\}$ and $\xrightarrow[n\to\infty]{\mathcal{L}|y\in\mathcal{E}_{0,k}}$ is defined similarly as $\xrightarrow[n\to\infty]{\mathcal{L}|Y\in\mathcal{E}_0}$.

3) Hence, for $x \in \mathbb{R}$, there exists a sequence $k_n \underset{n\to\infty}{\longrightarrow} \infty$ satisfying $k_n = o(n)$ as $n \to \infty$ so that:

$$\mathbb{P}\left(\sqrt{n}\left(\bar{\sigma}_n^2 - \sigma_0^2\right) \leqslant x \middle| \ell \leqslant \min_{i=1,\ldots,k_n} y_i \leqslant \max_{i=1,\ldots,k_n} y_i \leqslant u\right) \underset{n\to\infty}{\longrightarrow} \mathbb{P}\left(V \leqslant x\right)$$

with $V \sim \mathcal{N}(0, 2\sigma_0^4)$. Therefore, from Lemma A.7,

$$\mathbb{P}\left(\sqrt{n}\left(\bar{\sigma}_n^2 - \sigma_0^2\right) \leqslant x \middle| \, \ell \leqslant Y(x) \leqslant u, \ \forall x \in [0,1]^d\right) \underset{n\to\infty}{\longrightarrow} \mathbb{P}\left(V \leqslant x\right).$$

This concludes the proof. □

Theorem A.2 *For $\kappa = 1, 2$, we assume that Condition 7.2 holds. Under Condition 7.4 and Condition 7.3, the cMLE $\widehat{\sigma}_{n,c}^2$ of σ_0^2 defined in (7.12) is asymptotically Gaussian distributed. More precisely, for $\kappa = 0, 1, 2$,*

$$\sqrt{n}\left(\widehat{\sigma}_{n,c}^2 - \sigma_0^2\right) \underset{n\to+\infty}{\overset{\mathcal{L}|Y\in\mathcal{E}_\kappa}{\longrightarrow}} \mathcal{N}(0, 2\sigma_0^4).$$

Proof. We apply Lemma A.8 to the sequences of functions f_n, g_n and h_n defined by $f_n(\sigma^2) = \mathcal{L}_n(\sigma^2)$, $g_n(x) = A_n(\sigma^2)$, and $h_n(\sigma^2) = B_n(\sigma^2)$. Here we recall that for $\sigma^2 \in \Theta$,

$$A_n(\sigma^2) = -\log \mathbb{P}_{\sigma^2}\left(Y \in \mathcal{E}_0\right) \text{ and } B_n(\sigma^2) = \log \mathbb{P}_{\sigma^2}\left(Y \in \mathcal{E}_0 \middle| \, y\right).$$

In order to apply Lemma A.8, we need to check that the conditions (A.4) to (A.7) hold.

1) By (7.10), one has

$$\mathcal{L}_n(\sigma^2) = -\frac{n}{2}\log 2\pi - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\ln(|R_1|) - \frac{1}{2\sigma^2} y^\top R_1^{-1} y.$$

Now $y^\top R_1^{-1} y$ is the square of the norm of a Gaussian vector with variance-covariance matrix $\sigma_0^2 I_n$, where I_n stands for the identity matrix of dimension n. Thus one way write $y^\top R_1^{-1} y$ as the sum of the squares of n iid r.v.'s ε_i, where ε_i is Gaussian distributed with mean 0 and variance σ_0^2. We prove that (A.4) is satisfied. One may rewrite $\mathcal{L}_n(\sigma^2)$ as

$$\mathcal{L}_n(\sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log(|R_1|) + \frac{n}{2} f_{\sigma_0^2 + o_{\mathbb{P}}(1)}(\sigma^2), \tag{A.9}$$

where the $o_{\mathbb{P}}(1)$ above does not depend on σ^2 and f_a has been introduced in Appendix A.2.1. By a Taylor expansion and the definition of $\bar{\sigma}_n^2$, we have, with probability going to 1 as $n \to \infty$,

$$\begin{aligned}
\mathcal{L}_n(\sigma^2) - \mathcal{L}_n(\bar{\sigma}_n^2) &= (\sigma^2 - \bar{\sigma}_n^2)\mathcal{L}_n'(\bar{\sigma}_n^2) + \frac{1}{2}(\sigma^2 - \bar{\sigma}_n^2)^2 \mathcal{L}_n''(\tilde{\sigma}^2) \\
&= \frac{n}{4} f''_{\sigma_0^2 + o_{\mathbb{P}}(1)}(\tilde{\sigma}^2)(\sigma^2 - \bar{\sigma}_n^2)^2 \\
&= \frac{n}{4}\left(\frac{1}{\tilde{\sigma}^4} - 2\frac{\sigma_0^2 + o_{\mathbb{P}}(1)}{\tilde{\sigma}^6}\right)(\sigma^2 - \bar{\sigma}_n^2)^2,
\end{aligned}$$

with $\tilde{\sigma}^2$ in the interval with endpoints σ^2 and $\bar{\sigma}_n^2$. Hence, non-random constants $A > 0$ and $\delta > 0$ exist for which (A.4) is satisfied.

2) Second, let us prove that (A.5) holds with the previous $\delta > 0$ and for some $B > 0$. From (A.9), $2\mathcal{L}_n/n + \log(2\pi) + (1/n)\log(|R_1|)$ converges uniformly on $[\sigma_l^2, \sigma_u^2]$ as n goes

to infinity to $f_{\sigma_0^2}$. The function $f_{\sigma_0^2}$ attains its unique maximum at σ_0^2, which implies the result since $\bar{\sigma}_n^2$ converges to σ_0^2 in probability. Hence (A.5) hods.

3) Now we consider (A.6). Let us introduce the Gaussian process Y_r with mean function zero and covariance function k_1. Let $\sigma_1^2 \leqslant \sigma_2^2$. Then, one has:

$$\begin{aligned}\left|\exp\left\{-A_n(\sigma_1^2)\right\} - \exp\left\{-A_n(\sigma_2^2)\right\}\right| &= \left|\mathbb{P}(\sigma_1 Y_r \in \mathcal{E}_0) - \mathbb{P}(\sigma_2 Y_r \in \mathcal{E}_0)\right| \\ &\leqslant \mathbb{P}\left(\frac{u}{\sigma_2} \leqslant Y_r(x) \leqslant \frac{u}{\sigma_1},\ \forall x \in [0,1]^d\right) \\ &\quad + \mathbb{P}\left(\frac{\ell}{\sigma_2} \leqslant Y_r(x) \leqslant \frac{\ell}{\sigma_1},\ \forall x \in [0,1]^d\right) \\ &\leqslant c\left|\frac{1}{\sigma_1} - \frac{1}{\sigma_2}\right| \\ &\leqslant c|\sigma_2^2 - \sigma_1^2|\end{aligned}$$

by Tsirelson theorem in (Azaïs and Wschebor, 2009). Then, from Lemma 7.1, (A.6) holds.

4) We turn to

$$B_n(\sigma^2) = \log \mathbb{P}_{\sigma^2}(Y \in \mathcal{E}_0|y) = \log \mathbb{P}_{\sigma^2}(\ell \leqslant Y(x) \leqslant u,\ \forall x \in [0,1]^d|y).$$

Let $m_{n,y}$ and $\sigma^2 k_n$ be the conditional mean and covariance functions of Y given y, under the probability measure $\mathbb{P}_{\sigma^2}$. Using Borell-TIS inequality (Adler and Taylor, 2007), with Z_{n,σ^2} a Gaussian process with mean function zero and covariance function $\sigma^2 k_n$, we obtain

$$\begin{aligned}\mathbb{P}_{\sigma^2}(Y^* > u|y) &\leqslant \mathbb{P}_{\sigma^2}\left(Z^*_{n,\sigma^2} > u - \sup_{x\in[0,1]^d} m_{n,y}(x)|y\right) \\ &\leqslant \mathbb{P}_{\sigma^2}\left(Z^{**}_{n,\sigma^2} > u - \sup_{x\in[0,1]^d} m_{n,y}(x)|y\right) \\ &\leqslant \exp\left\{-\frac{\left(\left(u - \sup_{x\in[0,1]^d} m_{n,y}(x) - \mathbb{E}[Z^{**}_{n,\sigma^2}]\right)_+\right)^2}{2\sup_{x\in[0,1]^d}\mathbb{E}[Z_{n,\sigma^2}(x)^2]}\right\}. \qquad \text{(A.10)}\end{aligned}$$

But by Lemma A.9, $\sup_{\sigma^2\in[\sigma_l^2,\sigma_u^2]} \mathbb{E}[Z^{**}_{n,\sigma^2}] \to 0$ as $n \to +\infty$. Additionally, one can simply show that $\sup_{x\in[0,1]^d} \mathbb{E}[Z_{n,\sigma^2}(x)^2] = \sup_{x\in[0,1]^d} \sigma^2 k_n(x,x)$ goes to zero uniformly in $\sigma^2 \in [\sigma_l^2, \sigma_u^2]$ as $n \to \infty$. By (Bect et al., 2016, Proposition 2.8) and because the sequence of observation points is dense,

$$\sup_{x\in[0,1]^d} |m_{n,y}(x) - Y(x)| \xrightarrow[n\to+\infty]{\text{a.s.}} 0$$

from which we deduce that on $\{Y^* < u - \delta\}$, a.s.

$$\limsup_{n\to+\infty} \left(u - \sup_{x\in[0,1]^d} m_{n,y}(x) \right) \geqslant \delta.$$

Consequently, (A.10) leads to

$$\mathbb{1}_{\{Y^*<u-\delta\}} \sup_{\sigma^2\in[\sigma_l^2,\sigma_u^2]} \mathbb{P}_{\sigma^2}(Y^* > u|y) \xrightarrow[n\to+\infty]{\text{a.s.}} 0. \tag{A.11}$$

Similarly, taking $-Y$ instead of Y, one may prove easily that

$$\mathbb{1}_{\{Y_*>l+\delta\}} \sup_{\sigma^2\in[\sigma_l^2,\sigma_u^2]} \mathbb{P}_{\sigma^2}(Y_* < l|y) \xrightarrow[n\to+\infty]{\text{a.s.}} 0. \tag{A.12}$$

Then, we deduce that

$$\mathbb{1}_{\{\ell+\delta<Y(x)<u-\delta,\ \forall x\in[0,1]^d\}} \sup_{\sigma^2\in[\sigma_l^2,\sigma_u^2]} \mathbb{P}_{\sigma^2}(Y^* > u \text{ or } Y_* < \ell|y) \xrightarrow[n\to+\infty]{\text{a.s.}} 0. \tag{A.13}$$

Now let $\varepsilon > 0$, $\varepsilon' = 2|\log(1-\varepsilon)|$ and $\mathcal{E}_{0,\delta} := \{f \in \mathcal{C}([0,1]^d, \mathbb{R}) \quad \text{s.t. } \ell + \delta \leqslant f(x) \leqslant u - \delta,\ \forall x \in [0,1]^d\}$. We have:

$$\begin{aligned}
&\mathbb{P}\left(\sup_{\sigma^2\in[\sigma_l^2,\sigma_u^2]} \mathbb{P}_{\sigma^2}\left(Y^* > u \text{ or } Y_* < \ell|y\right) \geqslant \varepsilon,\ Y \in \mathcal{E}_{0,\delta} \right) \xrightarrow[n\to+\infty]{} 0 \\
&\qquad = \mathbb{P}\left(\inf_{\sigma^2\in[\sigma_l^2,\sigma_u^2]} B_n(\sigma^2) \leqslant -\varepsilon'/2,\ Y \in \mathcal{E}_{0,\delta} \right) \\
&\qquad = \mathbb{P}\left(\sup_{\sigma^2\in[\sigma_l^2,\sigma_u^2]} |B_n(\sigma^2)| \geqslant \varepsilon'/2,\ Y \in \mathcal{E}_{0,\delta} \right) \\
&\qquad \geqslant \mathbb{P}\left(\sup_{\sigma_1^2,\sigma_2^2\in[\sigma_l^2,\sigma_u^2]} |B_n(\sigma_1^2) - B_n(\sigma_2^2)| \geqslant \varepsilon',\ Y \in \mathcal{E}_{0,\delta} \right)
\end{aligned}$$

by the triangular inequality and (A.13). Therefore,

$$\begin{aligned}
&\mathbb{P}\left(\sup_{\sigma_1^2,\sigma_2^2\in[\sigma_l^2,\sigma_u^2]} |B_n(\sigma_1^2) - B_n(\sigma_2^2)| \geqslant \varepsilon',\ Y \in \mathcal{E}_0 \right) \\
=&\mathbb{P}\left(\sup_{\sigma_1^2,\sigma_2^2\in[\sigma_l^2,\sigma_u^2]} |B_n(\sigma_1^2) - B_n(\sigma_2^2)| \geqslant \varepsilon',\ Y \in \mathcal{E}_{0,\delta} \right) &\text{(A.14)} \\
&+ \mathbb{P}\left(\sup_{\sigma_1^2,\sigma_2^2\in[\sigma_l^2,\sigma_u^2]} |B_n(\sigma_1^2) - B_n(\sigma_2^2)| \geqslant \varepsilon',\ Y \in \mathcal{E}_0 \setminus \mathcal{E}_{0,\delta} \right). &\text{(A.15)}
\end{aligned}$$

As already shown, the term (A.14) converges to 0 as $n \to +\infty$ for any fixed $\delta > 0$. For (A.15), we have

$$\sup_{\substack{t_1,t_2\in R \\ t_1\neq t_2}} \frac{|\mathbb{P}_{\sigma_0^2}(Y^* \leqslant t_1) - \mathbb{P}_{\sigma_0^2}(Y^* \leqslant t_2)|}{|t_1 - t_2|} < +\infty.$$

This follows from Tsirelson theorem in (Azaïs and Wschebor, 2009). Hence for all $\varepsilon > 0$, there exists $\delta^* > 0$ such that,

$$\mathbb{P}_{\sigma_0^2}(Y^* \in [u - \delta^*, u]) \leqslant \varepsilon. \tag{A.16}$$

Similarly, for all $\varepsilon > 0$, there exists $\delta_* > 0$ such that,

$$\mathbb{P}_{\sigma_0^2}(Y_* \in [\ell + \delta_*, \ell]) \leqslant \varepsilon. \tag{A.17}$$

Taking $\delta = \min(\delta_*, \delta^*)$, we conclude the proof of (A.7).

5) Finally, we remark that with probabiliy going to one as $n \to \infty$, $\bar{\sigma}_n^2 = \arg\max_{\sigma^2 \in [\sigma_l^2, \sigma_u^2]} \mathcal{L}_n(\sigma^2)$. Hence, one may apply Lemma A.8 to obtain

$$\sqrt{n}|\widehat{\sigma}_{n,c}^2 - \bar{\sigma}_n^2| = o_{\mathbb{P}|Y \in \mathcal{E}_0}(1).$$

By Theorem 7.1 and Slutsky's lemma, we conclude the proof. □

References

(2014). *Criticality Safety in the Handling of Fissile Material: Specific Safety Guide*. International Atomic Energy Agency. IAEA Safety Standards Series No. SSG-27.

Abramowitz, M. and Stegun, I. A. (1964). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York.

Adams, R. P., Murray, I., and MacKay, D. J. (2009). Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities. In *International Conference on Machine Learning*, pages 9–16.

Adler, R. J. (1990). *An Introduction to Continuity, Extrema, and Related Topics for General Gaussian Processes*. IMS, Hayward, California.

Adler, R. J. and Taylor, J. E. (2007). *Random Fields and Geometry*. Springer Monographs in Mathematics. Springer, New York.

Azaïs, J.-M. and Wschebor, M. (2009). *Level Sets and Extrema of Random Processes and Fields*. Wiley & Sons, New York.

Azzimonti, D. (2018). *profExtrema: Compute and Visualize Profile Extrema Functions*. R package version 0.2.0.

Azzimonti, D., Ginsbourger, D., Rohmer, J., and Idier, D. (2019). Profile extrema for visualizing and quantifying uncertainties on excursion regions. Application to coastal flooding. *Technometrics*, 0(ja):1–26.

Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66:55–69.

Bachoc, F., Lagnoux, A., and López-Lopera, A. F. (2019). Maximum likelihood estimation for Gaussian processes under inequality constraints. *Electronic Journal of Statistics*, 13(2):2921–2969.

Baddeley, A., Gregori, P., Mahiques, J., Stoica, R., and Stoyan, D. (2006). *Case Studies in Spatial Point Process Modeling*. Lecture Notes in Statistics. Springer, New York.

Baddeley, A., Rubak, E., and Turner, R. (2015). *Spatial Point Patterns: Methodology and Applications with R*. Chapman & Hall/CRC Interdisciplinary Statistics. CRC Press, Boca Raton, FL.

Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483.

Bay, X., Grammont, L., and Maatouk, H. (2016). Generalization of the Kimeldorf-Wahba correspondence for constrained interpolation. *Electronic journal of statistics*, 10(1):1580–1595.

Bect, J., Bachoc, F., and Ginsbourger, D. (2016). A supermartingale approach to Gaussian process based sequential design of experiments. *ArXiv e-prints*.

Botev, Z. I. (2017). The normal law under linear restrictions: Simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B*, 79(1):125–148.

Brenner, S. and Scott, R. (2007). *The Mathematical Theory of Finite Element Methods*. Texts in Applied Mathematics. Springer, New York.

Brooks, S., Gelman, A., Jones, G., and Meng, X. (2011). *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, Boca Raton, FL, Boca Raton, FL.

Buja, A., Hastie, T., and Tibshirani, R. (1989). Linear smoothers and additive models. *The Annals of Statistics*, 17(2):453–510.

Cha, J. and Finkelstein, M. (2018). *Point Processes for Reliability Analysis: Shocks and Repairable Systems*. Springer Series in Reliability Engineering. Springer, New York.

Cousin, A., Maatouk, H., and Rullière, D. (2016). Kriging of financial term-structures. *European Journal of Operational Research*, 255(2):631–648.

Cox, D. R. (1955). Some statistical methods connected with series of events. *Journal of the Royal Statistical Society: Series B*, 17(2):129–164.

Creutzig, J., Müller-Gronbach, T., and Ritter, K. (2007). Free-knot spline approximation of stochastic processes. *Journal of Complexity*, 23(4):867–889.

Da Veiga, S. and Marrel, A. (2012). Gaussian process modeling with inequality constraints. *Annales de la faculté des sciences de Toulouse Mathématiques*, 21(3):529–555.

De Boor, C. (2001). *A Practical Guide to Splines*. Applied Mathematical Sciences. Springer, New York.

De Boor, C. (2002). Spline basics. In *Handbook of Computer Aided Geometric Design*, pages 141 – 163. North-Holland, Amsterdam.

De Loera, J., Rambau, J., and Santos, F. (2010). *Triangulations: Structures for Algorithms and Applications*. Algorithms and Computation in Mathematics. Springer, Berlin Heidelberg.

Delaunay, B. (1934). Sur la sphère vide. A la mémoire de Georges Voronoï. *Bulletin de l'Académie des Sciences de l'URSS*, (6):793–800.

Deville, Y., Ginsbourger, D., and Roustant, O. (2015). *kergp: Gaussian Process Laboratory*. R package version 0.2.0.

Diggle, P. J., Moraga, P., Rowlingson, B., and Taylor, B. M. (2013). Spatial and spatio-temporal log-Gaussian Cox processes: Extending the geostatistical paradigm. *Statistical Science*, 28(4):542–563.

Donner, C. and Opper, M. (2018). Efficient Bayesian inference of sigmoidal Gaussian Cox processes. *Journal of Machine Learning Research*, 19(67):1–34.

Du, J., Zhang, H., and Mandrekar, V. S. (2009). Fixed-domain asymptotic properties of tapered maximum likelihood estimators. *The Annals of Statistics*, 37(6A):3330–3361.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222.

Dupuy, D., Helbert, C., and Franco, J. (2015). DiceDesign and DiceEval: Two R packages for design and analysis of computer experiments. *Journal of Statistical Software*, 65(11):1–38.

Durrande, N., Adam, V., Bordeaux, L., Eleftheriadis, S., and Hensman, J. (2019). Banded matrix operators for Gaussian Markov models in the automatic differentiation era. In *International Conference on Artificial Intelligence and Statistics*, pages 2780–2789.

Durrande, N., Ginsbourger, D., and Roustant, O. (2012). Additive covariance kernels for high-dimensional Gaussian process modeling. *Annales de la Faculté de Sciences de Toulouse*, 21(3):481–499.

Duvenaud, D. K., Nickisch, H., and Rasmussen, C. E. (2011). Additive Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 226–234.

Fernandez, T., Rivera, N., and Teh, Y. W. (2016). Gaussian processes for survival analysis. In *Conference on Neural Information Processing Systems*, pages 5021–5029.

Fernex, F., Heulers, L., Jacquet, O., Miss, J., and Richet, Y. (2005). The MORET 4B Monte Carlo code - New features to treat complex criticality systems. In *M&C international conference on mathematics and computation supercomputing, reactor physics and nuclear and biological application, Avignon, France*, volume 59.

Flaxman, S., Wilson, A., Neill, D., Nickisch, H., and Smola, A. (2015). Fast Kronecker inference in Gaussian processes with non-Gaussian likelihoods. In *Proceedings of Machine Learning Research*, pages 607–616.

Flegal, J. M., Hughes, J., Vats, D., and Dai, N. (2017). *mcmcse: Monte Carlo Standard Errors for MCMC*. R package version 1.3-2.

Fruth, J. (2015). *Sensitivy analysis and graph-based methods for black-box functions with on application to sheet metal forming.* Theses, École Nationale Supérieure des Mines de Saint-Étienne.

Fruth, J., Roustant, O., and Kuhnt, S. (2014). Total interaction index: A variance-based sensitivity index for second-order interaction screening. *Journal of Statistical Planning and Inference*, 147:212–223.

Genton, M. G. (2001). Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2:299–312.

Genz, A. (1992). Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–150.

Geyer, C. J. (1992). Practical Markov Chain Monte Carlo. *Statistical Science*, 7(4):473–483.

Golchi, S., Bingham, D. R., Chipman, H., and Campbell, D. A. (2015). Monotone emulation of computer experiments. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):370–392.

Goldfarb, D. and Idnani, A. (1982). Dual and primal-dual methods for solving strictly convex quadratic programs. In *Numerical Analysis*, pages 226–239.

Goldman, R. (2003). B-spline approximation and the de Boor algorithm. In *Pyramid Algorithms*, pages 347–443. Elsevier, Amsterdam.

Gong, L. and Flegal, J. M. (2016). A practical sequential stopping rule for high-dimensional Markov Chain Monte Carlo. *Journal of Computational and Graphical Statistics*, 25(3):684–700.

Gunter, T., Lloyd, C. M., Osborne, M. A., and Roberts, S. J. (2014). Efficient Bayesian nonparametric modelling of structured point processes. In *Conference on Uncertainty in Artificial Intelligence*, pages 310–319.

Hastie, T. and Tibshirani, R. (1986). Generalized additive models. *Statistical Science*, 1(3):297–310.

Hensman, J., Durrande, N., and Solin, A. (2017). Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research*, 18(1):5537–5588.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence*, pages 282–290.

Hu, Y. (1993). An algorithm for data reduction using splines with free knots. *IMA Journal of Numerical Analysis*, 13(3):365–381.

Hyndman, R. J. and Fan, Y. (1996). Sample quantiles in statistical packages. *The American Statistician*, 50(4):361–365.

Ibragimov, I. and Rozanov, Y. (1978). *Gaussian Random Processes*. Springer, New York.

John, S. and Hensman, J. (2018). Large-scale Cox process inference using variational Fourier features. In *Proceedings of Machine Learning Research*, pages 2362–2370.

Johnson, S. G. The NLopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492.

Jupp, D. (1978). Approximation to data by splines with free knots. *SIAM Journal on Numerical Analysis*, 15(2):328–343.

Kallenberg, O. (2002). *Foundations of Modern Probability*. Probability and its Applications. Springer, New York.

Kaufman, C. and Shaby, B. (2013). The role of the range parameter for estimation and prediction in geostatistics. *Biometrika*, 100:473–484.

Kingman, J. (1992). *Poisson Processes*. Oxford Studies in Probability. Clarendon Press, New York.

Kobbelt, L. P. (2002). Multiresolution techniques. In *Handbook of Computer Aided Geometric Design*, pages 343–361. North-Holland, Amsterdam.

Kocijan, J. (2016). *Modelling and Control of Dynamic Systems using Gaussian Process Models*. Advances in Industrial Control. Springer, New York.

Kozachenko, Y., Pogorilyak, O., Rozora, I., and Tegza, A. (2016). Simulation of Cox random processes. In *Simulation of Stochastic Processes with Given Accuracy and Reliability*, pages 251–304. Elsevier, Amsterdam.

Lan, S. and Shahbaba, B. (2016). Sampling constrained probability distributions using spherical augmentation. In *Algorithmic Advances in Riemannian Geometry and Applications*, pages 25–71. Springer, New York.

Lasko, T. A. (2014). Efficient inference of Gaussian-process-modulated renewal processes with application to medical event data. *Uncertainty in artificial intelligence*, pages 469–476.

Lee, D. T. and Schachter, B. J. (1980). Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242.

Li, W. V. and Linde, W. (1999). Approximation, metric entropy and small ball estimates for Gaussian measures. *The Annals of Probability*, 27(3):1556–1578.

Lloyd, C. M., Gunter, T., Osborne, M. A., and Roberts, S. J. (2015). Variational inference for Gaussian process modulated Poisson processes. In *International Conference on Machine Learning*, pages 1814–1822.

Loh, W. L. (2005). Fixed-domain asymptotics for a subclass of Matérn-type Gaussian random fields. *The Annals of Statistics*, 33(5):2344–2394.

Loh, W. L. and Lam, T. K. (2000). Estimating structured correlation matrices in smooth Gaussian random field models. *The Annals of Statistics*, 28(3):880–904.

López-Lopera, A. F. (2019). *lineqGPR: Gaussian Process Regression Models with Linear Inequality Constraints*. R package version 0.0.4.

López-Lopera, A. F., Bachoc, F., Durrand e, N., Rohmer, J., Idier, D., and Roustant, O. (2019). Approximating Gaussian process emulators with linear inequality constraints and noisy observations via MC and MCMC. *arXiv e-prints*.

López-Lopera, A. F., Bachoc, F., Durrande, N., and Roustant, O. (2018). Finite-dimensional Gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1224–1255.

López-Lopera, A. F., John, S., and Durrande, N. (2019). Gaussian process modulated Cox processes under linear inequality constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 1997–2006.

Maatouk, H. (2015). *Correspondance entre régression par processus gaussien et splines d'interpolation sous contraintes linéaires de type inégalité. Théorie et application*. Theses, École Nationale Supérieure des Mines de Saint-Étienne.

Maatouk, H. (2017). Finite-dimensional approximation of Gaussian processes with inequality constraints. *ArXiv e-prints*.

Maatouk, H. and Bay, X. (2016). A new rejection sampling method for truncated multivariate Gaussian random variables restricted to convex sets. In *Monte Carlo and Quasi-Monte Carlo Methods*, pages 521–530. Springer, New York.

Maatouk, H. and Bay, X. (2017). Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences*, 49(5):557–582.

Maatouk, H., Roustant, O., and Richet, Y. (2015). Cross-validation estimations of hyper-parameters of Gaussian processes with inequality constraints. *Procedia Environmental Sciences*, 27:38–44.

Macdonald, I. (1998). *Symmetric Functions and Hall Polynomials*. Oxford University Press. USA.

Meyer, C. D. (2000). *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (2001). Log Gaussian Cox processes. *Scandinavian Journal of Statistics*, 25(3):451–482.

Møller, J. and Waagepetersen, R. P. (2004). *Statistical Inference and Simulation for Spatial Point Processes*. Chapman & Hall/CRC Monographs on Statistics and Applied Probability. CRC Press, Boca Raton, FL.

Muehlenstaedt, T., Roustant, O., Carraro, L., and Kuhnt, S. (2012). Data-driven Kriging models based on FANOVA-decomposition. *Statistics and Computing*, 22(3):723–738.

Munkres, J. (2018). *Elements Of Algebraic Topology*. CRC Press, Boca Raton, FL.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning Series)*. The MIT Press, Cambridge, MA.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer, New York.

Nickisch, H. and Rasmussen, C. E. (2008). Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9:2035–2078.

Okabe, A., Boots, B., and Sugihara, K. (1992). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley & Sons, New York.

Paciorek, C. J. and Schervish, M. J. (2004). Nonstationary covariance functions for Gaussian process regression. In *Conference on Neural Information Processing Systems*, pages 273–280.

Pakman, A. and Paninski, L. (2014). Exact Hamiltonian Monte Carlo for truncated multivariate Gaussians. *Journal of Computational and Graphical Statistics*, 23(2):518–542.

Pan, C. and Zhu, M. (2017). Group additive structure identification for kernel nonparametric regression. In *Advances in Neural Information Processing Systems*, pages 4907–4916.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge.

Pronzato, L. (2017). Minimax and maximin space-filling designs: Some properties and methods for construction. *Journal de la Société Française de Statistique*, 158(1):7–36.

Pronzato, L. and Müller, W. G. (2012). Design of computer experiments: filling and beyond. *Statistics and Computing*, 22(3):681–701.

Quiñonero Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, Cambridge, MA.

Ray, P., Pati, D., and Bhattacharya, A. (2019). Efficient Bayesian shape-restricted function estimation with constrained Gaussian process priors. *arXiv e-prints*.

Riihimäki, J. and Vehtari, A. (2010). Gaussian processes with monotonicity information. In *Journal of Machine Learning Research: Workshop and Conference Proceedings*, pages 645–652.

Ripley, B. D. (1977). Modelling spatial patterns. *Journal of the Royal Statistical Society: Series B*, 39(2):172–212.

Rohmer, J. and Idier, D. (2012). A meta-modelling strategy to identify the critical offshore conditions for coastal flooding. *Natural Hazards and Earth System Sciences*, 12(9):2943–2955.

Roustant, O., Ginsbourger, D., and Deville, Y. (2012). DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by Kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55.

Santner, T., Williams, B., and Notz, W. (2003a). *The Design and Analysis of Computer Experiments*. Springer, New York.

Santner, T., Williams, B., and Notz, W. (2003b). Space-filling designs for computer experiments. In *The Design and Analysis of Computer Experiments*, pages 121–161. Springer, New York.

Särkkä, S. (2013). *Bayesian Filtering and Smoothing*. Cambridge University Press, Cambridge.

Serfozo, R. (2009). Renewal and regenerative processes. In *Basics of Applied Stochastic Processes*, pages 99–167. Springer, New York.

Slassi, M. (2014). The optimal free knot spline approximation of stochastic differential equations with additive noise. *Journal of Computational and Applied Mathematics*, 261:62–71.

Sleeper, L. A. and Harrington, D. P. (1990). Regression splines in the Cox model with application to covariate effects in liver disease. *Journal of the American Statistical Association*, 85(412):941–949.

Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pages 1257–1264.

Stein, M. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Series in Statistics. Springer, New York.

Svanberg, K. (2002). A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, 12(2):555–573.

Taylor, J. and Benjamini, Y. (2016). *RestrictedMVN: Multivariate Normal Restricted by Affine Constraints*. R package version 1.0.

Teh, Y. W. and Rao, V. (2011). Gaussian process modulated renewal processes. In *Advances in Neural Information Processing Systems*, pages 2474–2482.

Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 567–574.

Turner, R. (2018). *deldir: Delaunay Triangulation and Dirichlet (Voronoi) Tessellation*. R package version 0.1-15.

Van der Vaart, A. W. (1998). *Asymptotic Statistics*, volume 3. Cambridge University Press, Cambridge.

Van der Wilk, M., Rasmussen, C. E., and Hensman, J. (2017). Convolutional Gaussian processes. In *Conference on Neural Information Processing Systems*, pages 2845–2854.

Vanhatalo, J. and Vehtari, A. (2007). Sparse log Gaussian processes via MCMC for spatial epidemiology. In *Proceedings of Machine Learning Research*, pages 73–89.

Vats, D., Flegal, J. M., and Jones, G. L. (2017). Multivariate output analysis for Markov Chain Monte Carlo. *ArXiv e-prints*.

Yadrenko, M. I. (1983). *Spectral Theory of Random Fields*. Translation series in mathematics and engineering. Optimization Software, New York.

Yannaros, N. (1988). On Cox processes and Gamma renewal processes. *Journal of Applied Probability*, 25(2):423–427.

Yannaros, N. (1994). Weibull renewal processes. *Annals of the Institute of Statistical Mathematics*, 46(4):641–648.

Ying, Z. (1993). Maximum likelihood estimation of parameters under a spatial sampling scheme. *The Annals of Statistics*, 21(3):1567–1590.

Zhang, H. (2004). Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association*, 99(465):250–261.

Zhang, J. and Lin, L. (2018). Bounded regression with Gaussian process projection. *arXiv e-prints*.

Zhou, S., Giulani, P., Piekarewicz, J., Bhattacharya, A., and Pati, D. (2019). Reexamining the proton-radius problem using constrained Gaussian processes. *Physical Review C*, 99:055202.

Zienkiewicz, O., Taylor, R., and Zhu, J. (2013). *The Finite Element Method: Its Basis and Fundamentals*. The Finite Element Method. Elsevier, Amsterdam.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT: 2019LYSEM020

Andrés Felipe LOPEZ-LOPERA

Gaussian Process Modelling under Inequality Constraints

Speciality: Applied Mathematics

Keywords: Bayesian Inference Methods, Gaussian Processes, Metamodelling under Inequality Constraints, Monte Carlo and Markov Chain Monte Carlo Methods.

Abstract:

Conditioning Gaussian processes (GPs) by inequality constraints gives more realistic models. This thesis focuses on the finite-dimensional approximation of GP models proposed by Maatouk (2015), which satisfies the constraints everywhere in the input space. Several contributions are provided. First, we study the use of Markov Chain Monte Carlo methods for truncated multinormals. They result in efficient sampling for linear inequality constraints. Second, we explore the extension of the model, previously limited up to three-dimensional spaces, to higher dimensions. The introduction of a noise effect allows us to go up to dimension five. We propose a sequential algorithm based on knot insertion, which concentrates the computational budget on the most active dimensions. We also explore the Delaunay triangulation as an alternative to tensorisation. Finally, we study the case of additive models in this context, theoretically and on problems involving hundreds of input variables. Third, we give theoretical results on inference under inequality constraints. The asymptotic consistency and normality of maximum likelihood estimators are established. The main methods throughout this manuscript are implemented in R language programming. They are applied to risk assessment problems in nuclear safety and coastal flooding, accounting for positivity and monotonicity constraints. As a by-product, we also show that the proposed GP approach provides an original framework for modelling Poisson processes with stochastic intensities.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2019LYSEM020

Andrés Felipe LOPEZ-LOPERA

Modélisation par Processus Gaussiens sous Contraintes d'Inégalité

Spécialité : Mathématiques Appliquées

Mots clefs : Méthodes d'Inférence Bayésienne, Processus Gaussiens, Métamodélisation sous Contraintes d'Inégalité, Méthodes de Monte Carlo et Monte Carlo par chaînes de Markov

Résumé :

Le conditionnement de Processus Gaussiens (PG) par des contraintes d'inégalité permet d'obtenir des modèles plus réalistes. Cette thèse s'intéresse au modèle de type PG proposé par Maatouk (2015), obtenu par approximation finie, qui garantit que les contraintes sont satisfaites dans tout l'espace. Plusieurs contributions sont apportées. Premièrement, nous étudions l'emploi de méthodes de Monte Carlo par chaînes de Markov pour des lois multinormales tronquées. Elles fournissent un échantillonnage efficace pour des contraintes d'inégalité linéaires. Deuxièmement, nous explorons l'extension du modèle, jusque-là limité à la dimension trois, à de plus grandes dimensions. Nous remarquons que l'introduction d'un bruit d'observations permet de monter à la dimension cinq. Nous proposons un algorithme d'insertion des nœuds, qui concentre le budget de calcul sur les dimensions les plus actives. Nous explorons aussi la triangulation de Delaunay comme alternative à la tensorisation. Enfin, nous étudions l'utilisation de modèles additifs dans ce contexte, théoriquement et sur des problèmes de plusieurs centaines de variables. Troisièmement, nous donnons des résultats théoriques sur l'inférence sous contraintes d'inégalité. La consistance et la normalité asymptotique d'estimateurs par maximum de vraisemblance sont établies. L'ensemble des travaux a fait l'objet d'un développement logiciel en R. Ils sont appliqués à des problèmes de gestion des risques en sûreté nucléaire et inondations côtières, avec des contraintes de positivité et monotonie. Comme ouverture, nous montrons que la méthodologie fournit un cadre original pour l'étude de processus de Poisson d'intensité stochastique.