Sampling the solutions of differential systems

Christian Paul Chan Shio

UNIVERSITÉ DE NICE-SOPHIA ANTIPOLIS - UFR Sciences

École Doctorale de Sciences Fondamentales et Appliquées

THÈSE

pour obtenir le titre de

Docteur en Sciences

de l'UNIVERSITÉ de Nice-Sophia Antipolis

Discipline: **Mathematiques**

présentée et soutenue par

Christian Paul CHAN SHIO

Échantillonner les solutions de systèmes différentiels

Sampling the solutions of differential systems

Thèse dirigée par Francine DIENER
soutenue le 11 decembre 2014

Jury:

M. Eduardo MENDOZA	Professeur, Université des Philippines Diliman	Rapporteur
M. Gauthier SALLET	Professeur Émérite, Université de Lorraine	Rapporteur
M. Augustin FRUCHARD	Professeur, Université de Haute-Alsace	Examinateur
Mme. Francine DIENER	Professeur, Université de Nice-Sophia Antipolis	Directeur de Thèse

Acknowledgments

First giving thanks to the almighty God and saviour Jesus Christ, without whom nothing would be possible.

There have been a large number of people who have been instrumental in the success of my doctorate. The biggest and most heartfelt thanks go to my supervisor, Francine Diener, without whom I would not have probably reached this point. Thank you for your support and encouragement, especially during the times when I was down and did not feel my work was good enough. Thank you for your patience and generosity with your time to this inexperienced researcher. I have learned a lot from all the discussions we had over the past three years and I know I have emerged from this experience as a better mathematician because of you.

I am grateful to my reporters and jury members, Gauthier Sallet, Eduardo Mendoza and Augustin Fruchard for their helpful comments, corrections, and suggestions, all of which have provided substantial improvement to my work. Thanks also to Marc Diener for the discussions and for reading and improving parts of my work.

I give special acknowledgment to the scholarship programme of European Union, Erasmus Mundus Mobility with Asia (EMMA), which offered me the financial support for doing this PhD research. My special thanks also to all EMMA administrative staffs who were there whenever I was in need. Also, our local EMMA coordinators, Dr. Jumela Sarmiento and Dr. Reggie Marcelo back home in Ateneo for encouraging me and supporting my application.

This was the first time I have been away from home for an extended period of time. I thank my parents who have been very supportive of my decision to study abroad. Also, the company of several friends and co-students, not only in France but even elsewhere in Europe were instrumental in keeping me sane and prevented me from being homesick. In particular, I would like to thank Stephanie, Bridge, EC, Jas, Joy, Bien, Huong, Aparna, Sijie, Bugs, Shine, Cherry, Joebell. Thank you also to my churchmates, especially Arbie, Choy, Mariz, Fides, Emily, Beth, Rommel, for their prayers and support.

Introduction en français

Beaucoup de phénomènes naturels, par exemple en biologie, sont modélisés par des systèmes d'équations différentielles. Ces modèles contiennent habituellement de nombreux coefficients que nous pouvons souhaiter ajuster à des données observées. Toutefois, en raison des erreurs de mesure de la variabilité des conditions expérimentales et d'autres incertitudes, il peut se révéler impossible de leur assigner une valeur précise. Une option plus réaliste est de considérer ces coefficients comme des *variables aléatoires*, et donc, de modéliser les phénomènes au moyen de systèmes differentiels à équations différentielles avec des *coefficients aléatoires*.

Motivation et problèmes abordés

Cette thèse vise à construire des outils efficaces pour les non-mathématiciens qui souhaitent comprendre et appliquer des systèmes différentiels à coefficients aléatoires. De fait, notre contribution se situe plus au niveau d'aides concrètes à l'utilisation de ces modèles et d'exemples instructifs utiles à leur compréhension plutôt qu'au niveau de résultats mathématiques généraux. Je me suis efforcé d'étudier les systèmes différentiels à coefficients aléatoires au moyen d'une approche de simulation. Ce qui place notre étude au croisement de celles des systèmes différentiels, des probabilités et des statistiques.

La première partie de ce travail examine la loi à un instant t^* fixé de la solution $y(t;\theta)$, issue d'une condition initiale donnée, d'une equation differentelle $y' = g(y;\theta)$. Il existe de nombreux scénarios pratiques où cette connaissance serait très utile. Par exemple, en pharmacocinétique, il est important de connaître la quantité d'un certain agent pathogène restante, plusieurs heures après qu'un certain médicament ait été adminstré. Cependant, il peut y avoir une certaine variabilité de l'effet du médicament en fonction des caractéristiques des individus. La connaissance de la distribution des solutions à l'instant t^* peut permettre au médecin de mieux comprendre les mécanismes d'assimilation du médicament.

D'autre part, étant donné que l'estimation des paramètres est essentielle pour tout modèle mathématique, il est également important de développer des méthodes pour estimer les paramètres d'un système d'équations différentielles à partir de la connaisance une solution en un nombre fini d'instants. Ce problème est traité dans la deuxième partie de ce travail ; comment estimer les paramètres

d'un système de différentiel connaissant les valeurs de la solution sur un ensemble fini d'instants. C'est un problème classique lequel les méthodes disponibles sont nombreuses. Cependant, la plupart de ces méthodes sont des méthodes déterministes qui fournissent seulement une estimation ponctuelle des paramètres. Dans notre approche, nous proposons plusieurs méthodes alternatives permettant de donner une *distribution* de valeurs des coefficients susceptibles d'être les bons coefficients, plutôt qu'une estimation ponctuelle. Cela nous permet non seulement de prendre en considération les erreurs et les incertitudes sur les données, mais aussi, de fournir au besoin une estimation ponctuelle.

Plan de la thèse

Cette thèse est structurée en quatre chapitres et une annexe.

Le chapitre 1 donne une examen de plusieurs resultats de probabilité et d'équations différentielles qui sont nécessaires pour le reste de la thèse. Les concepts de probabilité abordés comprennent des résultats de convergence, des transformations de lois, et des resultats sur les chaînes de Markov. Pour les équations différentielles, je rappelle quelque résultats de différentiabilité de la solution par rapport aux conditions initiales et aux coefficients et je présente les principaux exemples utilisés par la suite.

Le chapitre 2 se propose de décrire la loi à un instant donné t^* des solutions d'un système de équations différentielles $y' = g(y;\theta)$ où θ sont des coefficients qui l'on suppose aléatoires. La distribution de la variable aléatoire $y(t^*)$ se révèle être beaucoup plus difficile à détermine que ce que l'on peut le penser d'abord. Notre contribution apporte des réponses partielles. Pour l'étude de la distribution au temps t^*, on a, en effet, besoin de prendre au moins deux choses en considération : d'une part, que l'on peut rencontrer lois n'ayant aucun moments finis, et d'autre part, que pour certains systèmes différentiels, le problème de l'explosion en temps fini représente un obstacle pour les simulations. En outre, nous montrons sur un exemple qu'un développement de la variable aléatoire $y(t^*)$ en chaos polynomial peut donner une bonne approximation de cette loi, au moins dans certains cas simples.

Á partir du chapitre 3, nous attachons notre attention à l'estimation des coefficients d'un système d'équations différentielles lorsqu'on connait les valeurs d'une solution en un petit nombre d'instants. Nous présentons d'abord une méthode de Monte Carlo simple, la méthode de rejet, qui permet de construire un échantillon de valeurs des coefficients θ compatibles avec les données. Nous offrons un aperçu des propriétés de cette méthode, indiquons comment choisir les différents paramètres qui doivent être choisis lors de la mise en œuvre de la méthode. Nous montrons également qu'il est possible d'améliorer l'efficacité de cette méthode en utilisant une nouvelle approche en deux étapes que nous appelons méthode de rejet séquentiel.

Le dernier chapitre (chapitre 4) présente une généralisation du chapitre précédent, où l'on remplace la méthode de Monte Carlo de base par des algorithmes plus élaborés, la methode MCMC dite « Markov Chain Monte Carlo » et l'algorithme de Monte Carlo séquentiel. Comme dans le chapitre précédent, notre contribution consiste principalement à expliquer sur des exemples comment mettre en œuvre ces algorithmes, mais aussi à fournir des indication sur la meilleure manière de sélectionner les différents paramètres nécessaires à cette mise en œuvre afin d'obtenir des résultats intéressants.

Une grande partie de notre recherche a consisté à réaliser des expériences avec Scilab. L'annexe fournit le code source de certains des programmes utilisés pour produire les résultats et les figures dans le texte. Comme pour le choix des expériences discutées du texte, plutôt que de fournir une liste exhaustive, cette annexe fournir un aperçu de la variété des programmes qui ont été préparés au cours de ce travail. Ces programmes et quelques autres sont disponibles sous forme de fichiers exécutables (.sce) sur ma page web http://math.unice.fr/~chanshio.

Conclusion en français

Dans cette thèse, nous avons étudié les systèmes differentiels à coefficients aléatoires au moyen de simulations. Nous avons pu voir que la loi des solutions de tel système à un instant t^* donné étant souvent impossible à calculer explicitement, même dans les cas les plus simples. Il est nécessaire de recourir à des simulations de Monte Carlo pour l'étude de cette loi. Cependant, simuler cette loi n'est pas toujours possible. Dans le cas d'une équation de Riccati dont les solutions explosent en temps fini, nous avons vu qu'une compactification de l'espace permet de représenter néanmoins son histogramme. Une autre possibilité envisagée est de calculer une approximation de cette loi au moyen d'un développement en chaos polynomial.

Concernant l'estimation des coefficients d'un système d'équations différentielles qui sont compatibles avec une trajectoire donnée, nous avons décrit l'algorithme de rejet qui produit une distribution de probabilité des meilleurs coefficients possibles. Cela fournit non seulement la possibilité de prendre en considération les erreurs et les incertitudes sur les données, mais aussi, de fournir au besoin une estimation ponctuelle. En supposant que la valeur réelle des coefficients existe, nous l'avons vu à travers plusieurs exemples que pour des cas de dimension faible et un assez petit seuil ϵ, on peut ainsi obtenir une distribution a posteriori qui permet de calculer bonnes estimations des coefficients θ. Toutefois, lorsque le nombre de coefficients augmente ou lorsque ϵ est trop petit, nous avons vu que pourcentage d'éléments acceptés diminue (et donc la taille de l'échantillon), et cela conduit alors à des estimations moins précises. C'est pourquoi nous avons proposons une méthode d'échantillonnage qui utilise les connaissances acquises au cours des premières itérations. Ceci permet d'augmenter le taux d'acceptation. En utilisant des methodes alternatives comme la méthode MCMC appelée Markov Chain Monte Carlo et la methode Monte Carlo séquentielle, nous avonsvu enfin qu'on peut aussi augmenter ce taux d'acceptation et diminuer les risques de rester bloquer dans un minimum local de la distance.

Résumé long en français

Beaucoup de phénomènes naturels, par exemple en biologie, sont modélisés par des systèmes d'équations différentielles. Ces modèles contiennent habituellement de nombreux coefficients qu'on peut souhaiter ajuster à des données observées. Mais en raison des erreurs de mesures, de la variabilité des conditions expérimentales et d'autres incertitudes, il peut se révéler impossible, et bien souvent illusoire, de leur assigner une valeur précise. Une option plus réaliste est de considérer ces coefficients comme des *variables aléatoires* et donc de modéliser les phénomènes étudiés au moyen de *systèmes différentiels à coefficients aléatoires*. Comme de tels modèles sont souvent utilisés par des non-mathématiciens, cette thèse a pour origine le souhait de construire des outils efficaces pour de tels scientifiques afin de leur permettre de mieux comprendre et d'utiliser plus facilement ces systèmes à coefficients aléatoires. De fait, notre contribution se situe plutôt au niveau d'aides concrètes à l'utilisation de ces modèles et d'exemples instructifs utiles à leur compréhension plutôt qu'au niveau de résultats mathématiques généraux. Principalement, l'étude de ces systèmes d'équations différentiels à coefficients aléatoires est faites ici au moyen de simulations, ce qui place notre étude au croisement du domaine des systèmes différentiels, de celui des probabilités et celui des statistiques. Toutes les simulations et intégrations numériques de systèmes différentiels ont été faites en utilisant Scilab.

La première partie de ce travail, intitulée « Loi des solutions à l'instant t^* »[1] étudie la loi à un instant $t^* > 0$ fixé de la solution $y(t;\theta)$, issue d'une condition initiale donnée, d'une équation différentielle $y' = g(y;\theta)$. La quantité $y(t^*;\theta)$ est une variable aléatoire qui est simplement l'image à l'instant t^* de la loi des coefficients aléatoires θ par la dynamique associée à l'équation différentielle. Il y a beaucoup de situations où une bonne connaissance de la loi de $y(t^*;\theta)$ peut être utile. Par exemple en pharmacocinétique, il est important de connaitre la quantité de certains pathogènes qui subsiste plusieurs heures après l'administration d'un médicament. Mais il y a sans doute de la variabilité dans les effets du médicament selon les caractéristiques des patients auxquels il est administré.

[1] qui suit un chapitre de préliminaires où ont été regroupés les principaux résultats classiques utilisés dans la thèse, tels que la dépendance des solutions d'un système différentiel par rapport à ses coefficients ou la loi des grands nombres

La connaissance de la distribution des solutions à l'instant t^* peut permettre au praticien de mieux comprendre les mécanismes d'assimilation du médicament.

L'étude de la loi de $y(t^*;\theta)$ n'est pas aussi simple qu'il peut sembler au premier abord. En général, on ne peut pas exprimer cette loi de probabilité comme l'image des coefficients aléatoires θ par une fonction mathématique connue (sauf si l'équation différentielle est intégrable par quadrature). Il est donc naturel d'avoir recours à *une approche par simulation* pour obtenir un histogramme permettant de se faire une idée de cette loi. Mais nous montrons que cette approche par simulation rencontre néanmoins au moins deux problèmes. Le premier que nous présentons dans le cas de l'équation différentielle la plus simple, linéaire et dépendant d'un unique coefficient aléatoire, est lié à la simulation de l'inverse d'une gaussienne, qui est l'exemple le plus simple de loi de probabilité n'ayant aucun moment fini. L'histogramme de la loi simulée ne laisse apparaitre, dans la fenêtre où on le représente qu'une partie seulement de ses valeurs, l'autre partie étant constituée de valeurs si « dispersées » à l'infini qu'elles en deviennent invisibles à distance "finie". Ce phénomène persiste même si l'on élargit la fenêtre parce que les valeurs hors fenêtre gardent toujours une mesure substantielle (d'où l'absence de moments finis pour cette loi). Nous montrons comment contourner ce problème en concentrant aux deux extrémités de l'histogramme les valeurs qui tombent hors de la fenêtre. Mais notre étude se limite à un seul exemple et la solution proposée propose seulement un moyen de représenter graphiquement la loi. Le second problème apparaît lorsque parmi les solutions simulées certaines explosent en temps fini. Nous étudions ce problème sur l'exemple d'une équation de Riccati, toujours dans le cas simple où un seul coefficient est aléatoire. La simulation d'un échantillon de valeurs de $y(t^*;\theta)$, devient tout simplement impossible dans ce cas, même si sa taille est petite et même si l'instant $y(t^*)$ choisi n'est pas trop grand car au moins une solution simulée va exploser avant l'instant t^*. Pour surmonter cette difficulté, nous proposons une compactification de l'ensemble des solutions de l'équation de Riccati par une transformation du type $y \mapsto Y = \frac{1}{y}$ qui permet de suivre la solution qui explose dans la second carte.

Les deux problèmes précédents montrent des difficultés rencontrées dans l'approche par simulation et donc l'utilité d'une approche alternative pour l'étude de la loi de la variable aléatoire $y(t^*;\theta)$. Nous présentons une telle approche alternative qui consiste à calculer une approximation de la loi de probabilité considérée au moyen d'un développement appelé *chaos polynomial*. L'approximation se calcule en projetant la variable aléatoire sur une base orthogonale de variables aléatoires qui est construite à partir d'une variable aléatoire donnée et de ses images par une famille de polynomes orthogonaux. On peut s'assurer que ce développement converge en probabilité, et même quelquefois en norme L^2, vers la variable étudiée. Toujours en choisissant des examples simples nous

vérifions que ces développements, même tronqués à un petit nombre de termes, peuvent fournir de bonnes approximations de la loi étudiée.

La seconde et la troisième partie de ce travail sont consacrées à l'étude de méthodes d'estimation des coefficients d'un système différentiel $y' = g(y;\theta)$, y étant cette fois de dimension l, lorsqu'on connait une solution « discrète », c'est-à-dire une solution $y(t;\theta)$ en un nombre fini d'instants $(t_0, t_1, \ldots, t_k)$. C'est un problème classique et les méthodes pour le résoudre pourraient remplir des livres entiers. Mais la plupart de ces méthodes sont *déterministes* en ce sens qu'elles fournissent une valeur unique des paramètres θ et, la plupart du temps, peu ou pas d'indication sur la précision de la valeur fournie. Si l'on pense à améliorer cette estimation ponctuelle en calculant des intervalles de confiance pour ces estimations, ce qui d'ailleurs n'est pas facile en général, on réalise que ces intervalles sont déjà une façon de remplacer l'estimation ponctuelle par le calcul d'un ensemble de valeurs possibles. Nous poursuivons dans cette direction en choisissant d'explorer ici des méthodes de type Monté Carlo qui conduiront au calcul d'une loi de probabilité pour les coefficients plutôt que de leur estimation ponctuelle. Cela permet à la fois de prendre en compte les erreurs et incertitudes des données observées mais aussi d'en déduire au besoin une estimation ponctuelle. La première méthode, dont l'étude fait l'objet de cette deuxième partie intitulée « Estimer les coefficients : une première approche », concerne la méthode dite *méthode de rejet*. Elle consiste à choisir un échantillon $\theta_1, \theta_2, \ldots, \theta_N$ de taille N, dont la loi, dite *prior*, a été choisi convenablement, puis de ne garder les θ obtenus seulement si la distance ρ de la solution, correspondant à cette valeur θ, calculée aux instants $(t_0, t_1, \ldots, t_k)$ avec celles de la solution discrète de référence n'excède pas un seuil ϵ choisi. Un choix naturel pour le prior est celui d'une loi uniforme sur le produit cartésien d'intervalles qui représentent chacun un intervalle de valeurs possibles pour l'un des coefficient. Bien que cette méthode soit plutôt naïve et facile à mettre en œuvre, elle donne souvent des résultats satisfaisants requiert de nombreux choix préalables, notamment celui du prior, de la distance, du seuil et de la taille de l'échantillon. Et lorsque ces choix ne sont pas fait de façon convenable, la méthode devient inopérante. C'est la raison qui a motivée l'étude détaillée de ses propriétés.

Les échantillons que la méthode de rejet produit ont d'intéressantes propriétés. Lorsque le seuil ϵ choisi est assez petit, ils sont contenus dans un ellipsoïde dont on peut calculer le demi axe principal en fonction des valeurs propres de la Hessienne de $\rho(\theta)$ évaluée au point θ_0 qui réalise le minimum de cette distance. Le pourcentage de valeurs de l'échantillon qui ne sont pas rejetés, est donc, pour ϵ assez petit, facile à estimer et indépendant de la taille N de l'échantillon, si ce n'est que sa variabilité sera plus faible lorsque N est plus grand. Comme ce

pourcentage décroit lorsque ϵ décroit ou que le nombre de coefficients inconnus croit, il est important, si l'on ne veut pas produire des échantillons presque vides ou même vides, de trouver le moyen d'assurer un taux d'acceptation suffisamment élevé. Cet objectif est facile à atteindre si l'on parvient à choisir le prior suffisamment « concentré » dans la "bonne" région. C'est dans ce but que nous proposons une approche nouvelle de la méthode de rejet, méthode en deux étapes, que nous appelons *méthode de rejet séquentielle*. Elle consiste à construire un premier (et petit) échantillon selon la méthode de rejet classique puis à poursuivre l'èchantillonnage mais cette fois en utilisant un prior de loi gaussienne ayant comme espérance et covariance, la moyenne et la covariance empirique calculées sur l'échantillon obtenu à la première étape. On peut vérifier sur des exemples que le taux d'acceptation de la méthode de rejet séquentielle est nettement plus grand que celui de la méthode de rejet classique et qu'en fait ce taux devient, sous certaines conditions, indépendant de ϵ.

Il existe bien sûr des méthodes plus élaborées que la méthode de rejet, notamment la méthode MCMC dite *Monté Carlo Markov Chain* et la méthode de Monté Carlo séquentielle. Leur étude fait l'objet de la dernière partie de cette thèse. Elles sont, l'une et l'autre des améliorations de la méthode de rejet et permettent notamment, tout comme la méthode de rejet séquentielle introduite dans la partie précédente, d'augmenter le taux d'acceptation. On obtiendra ainsi, pour une taille de l'échantillon de départ N donné, soit sensiblement plus de points dans l'échantillon retenu pour un ϵ donné et donc une meilleure connaissance de la loi des coefficients (et une meilleurs estimation ponctuelle au besoin), soit, un échantillon de même taille mais constitué de points acceptés pour un ϵ plus petit, donc des points plus précis.

L'algorithme MCMC utilise, à chaque étape de la construction de l'échantillon, des petits mouvements locaux autours de la valeur précédente tirant partie du fait que la distance ρ est une fonction continue de θ puisque les solutions du système différentiel le sont également. Une fois trouvé un premier θ non rejeté, on va ainsi choisir le point suivant « à proximité ». En incorporant alors à la condition d'acceptation, une contrainte supplémentaire dite *condition de Metropolis* on peut appliquer la théorie des chaines de Markov pour établir que l'échantillon ainsi construit aura la même loi que celle des échantillons obtenus par la méthode de rejet. Après avoir présenté ce résultat, on applique la méthode MCMC à différents exemples, on examine les propriétés des échantillons ainsi obtenus et on les compare à ceux de la méthode de rejet.

Parmi les problèmes que l'on peut rencontrer en appliquant la méthode MCMC, figure celui d'échantillons qui ne parviennent pas à s'étaler dans l'ensemble de la région explorée parce qu'ils restent bloqués à proximité d'un minimum local de ρ qui n'est pas nécessairement le minimum global dont on cherche

à approcher. On peut éviter ce problème (ou réduire le risque de le subir), en utilisant la méthode de Monté Carlo séquentielle. L'idée est d'améliorer le prior utilisé pour échantillonner, au fur et à mesure de la construction de l'échantillon, en choisissant une suite décroissante de valeurs de ϵ, dont la plus petite est la valeur choisie initialement. Pour chaque ϵ, un échantillon est construit par un choix aléatoire au sein de l'échantillon précédent dans lequel des poids ont été attribués à chaque élément, modifiant ainsi le prior du prochain échantillon, et un mouvement local autours du point obtenu, similaire à celui qui est fait dans l'algorithme MCMC, est effectué. Nous montrons sur quelques exemples comment la méthode peut être mise en œuvre puis nous expliquons pourquoi la distribution de l'échantillon final ainsi construit a bien la même distribution que les échantillons obtenus par la méthode de rejet.

Introduction

Many natural phenomena, for example in biology, are modeled using systems of differential equations. These models usually involve coefficients that are computed from observed data. However, due to measurement error in the data, variability in experimental conditions, or other uncertainties, it may not be possible to assign a specific value to the coefficients of the differential equations. A more appropriate way to specify these coefficients might then be to consider them as *random variables*, and thus, to model the phenomenon using a system of differential equations with *random coefficients*.

Motivation and Problems Addressed

This thesis stems from the desire to build efficient tools for non-mathematicians who wish to understand and apply systems of differential systems with random coefficients. As such, our contribution is more on ideas, practical usage, and interesting examples instead of general mathematical results. Here, I have attempted to study systems of differential equations with random coefficients using a simulation approach. This places us therefore in the crossroads of the fields of differential equations, probability, and statistics.

In the first part, given the distribution of the coefficients in a system of differential equations with random coefficients, we wish to look at the resulting distribution of the solution at some fixed time t^*. There are many practical scenarios where this knowledge would be very useful. For example, in pharmacokinetics, it is important to know the quantity of a certain pathogen remaining several hours after a certain drug is administered. However, there may be some variability in the effect of the drug depending on the characteristics of the individuals. Knowing the distribution of the solution at that time t^* can give the medical practitioner a better understanding of the assimilation mechanisms of the drug.

On the other hand, since coefficient estimation is central to any mathematical modeling, it is also important to further develop methods to estimate the parameters of a system of differential equations based on the knowledge of a discrete trajectory. This problem is addressed in the second part of this work; that is, how to "best" estimate the parameters of a differential system, given only the values of its solution for a finite set of time points. This is a popular problem

where the various available methods easily cover several books. However, most of these methods are deterministic methods which provide just a point estimate of the parameters. In our approach, we propose several variations of a method to give a *distribution* of points which are likely to be the true coefficients, instead of just a single point estimate. This allows us to not only take into consideration the errors and uncertainties in the known data, but at the same time, to provide a point estimate if necessary.

Outline of the Thesis

This thesis is structured into four chapters and an appendix as follows:

Chapter 1 provides a review of several concepts in probability and differential equations which are necessary for the remainder of the work. The probability concepts discussed include convergence results, transformations of laws, and Markov chains. For differential equations, I review the necessary results on differentiability of the solution with respect to initial conditions and coefficients.

Chapter 2 is mainly concerned with the problem of describing the law at a fixed time $t^\star$ of a system of differential equations $y' = g(y;\theta)$ where θ are coefficients which are random variables. This produces a random variable $y(t^\star)$ whose distribution turns out to be much more difficult than what one would initially think. Our contribution consists of partial answers to this problem. In particular, we shall show that when studying the distribution at time $t^\star$, one needs to take at least two things into consideration: first, that one may encounter laws without finite moments, and second, that for certain differential systems, the problem of explosion at finite time can be encountered and represent an obstacle for simulations. In addition, we show on an example that an expansion of the required random variable $y(t^\star)$ using polynomial chaos may give a good approximation and thus provide a tool to solve the problem, at least in the simplest cases.

Beginning with Chapter 3, our focus shifts to that of determining the best distribution of the coefficients in a system of differential equations given some data $\bar{y}$. We first introduce a simple Monte Carlo sampling method, the rejection method, to obtain a collection of points that are "close" to $\bar{y}$. We provide some insights on the properties of this method, as well as interesting advice on how to choose the different parameters that need to be chosen when implementing the method. We also show that it is possible to improve the efficiency of this method by using a new two-step approach which we call sequential rejection sampling.

The final chapter (Chapter 4) is an extension of the previous chapter, where we replace the basic Monte Carlo sampling method with more sophisticated tools. These tools are based on the Markov chain Monte Carlo and Sequential Monte Carlo algorithms in statistics. As in the previous chapter, our contribution centers on not only providing a friendly introduction to the algorithms, but also

on some commentary on how to select the different parameters of these methods for getting interesting results.

A large portion of our research involved implementing and performing experiments in Scilab. The appendix provides the source code of some of the programs used to produce the results and figures in the text. As in the choice of experiments discussed in the text, rather than being an extensive list, this appendix is designed to give the reader a glimpse of the variety of programs which were prepared during the course of this work. These programs and a few others are available as .sce files in my web page http://math.unice.fr/~chanshio.

Contents

Acknowledgments 3

Introduction en français 5

Conclusion en français 9

Résumé long en français 11

Introduction 17

Chapter 1. Preliminaries 23
1.1. Probability 23
1.2. Differential Equations 33
1.3. Higher-dimensional ellipsoids 35
1.4. Five Main Examples 37

Chapter 2. Law of the Solution at time t^* of a Differential Equation with Random Coefficients 41
2.1. An example in the linear case 41
2.2. An example in the Riccati case 46
2.3. Polynomial Chaos 51

Chapter 3. Estimating coefficients of systems of differential equations: a first approach 63
3.1. An example using a logistic model 63
3.2. An overview of ODE coefficient estimation methods 65
3.3. The rejection sampling algorithm 67
3.4. An analysis of the rejection sample 69
3.5. Improving the method 83
3.6. Application to perturbed model data 90

Chapter 4. Estimating coefficients of systems of differential equations: further approaches 99
4.1. A Markov chain Monte Carlo method 99
4.2. A Sequential Monte Carlo method 108

Conclusion 115

Bibliography 117

Appendix A. Scilab code 121

CHAPTER 1

Preliminaries

In this chapter, we first provide a brief review of the tools in probability theory (Section 1.1) and differential equations (Section 1.2) that will be necessary in the upcoming chapters. In Section 1.3, a simple formula of the volume of an ellipsoid is recalled, and finally, we shall give in the last section (Section 1.4) a short introduction to the examples of differential systems that we will use to illustrate our results in the next chapters.

1.1. Probability

1.1.1. Convergence results. Unless stated otherwise, we assume that all the random variables defined within a sequence $\{X_n\}$ are defined in the same probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Also, all expectations are to be taken over the probability measure $\mathbb{P}$.

Definition 1.1. *Let $\{X_n\}$ be a sequence of random variables. We say that $\{X_n\}$ converges in probability to the random variable X if for every $\epsilon > 0$,*

$$\lim_{n\to\infty} \mathbb{P}(|X_n - X| \geq \epsilon) = 0.$$

We denote convergence in probability of $\{X_n\}$ towards X by a right arrow with a $\mathcal{P}$ on top; that is, $X_n \xrightarrow{\mathcal{P}} X$.

Definition 1.2. *Let $\{X_n\}$ be a sequence of random variables. We say that $\{X_n\}$ converges in distribution to the random variable X if for almost all x,*

$$\lim_{n\to\infty} F_n(x) = F_X(x),$$

where F_n and F_X are the cumulative distribution functions of X_n and X, respectively.

We denote convergence in distribution of $\{X_n\}$ towards X by a right arrow with a $\mathcal{D}$ on top; that is, $X_n \xrightarrow{\mathcal{D}} X$.

While convergence in probability always implies convergence in distribution, the converse is not always true. However, if $\{X_n\}$ converges in distribution to a *constant*, then it can be shown that the sequence also converges in probability to that same constant.

DEFINITION 1.3. *Let X be a random variable. The characteristic function of X is defined by $\varphi(t) = \varphi_X(t) = \mathbb{E}(e^{itX})$.*

The following proposition, which follows directly from the definition, provides two important properties of the characteristic function.

PROPOSITION 1.4. *Let X and Y be independent random variables and a any real number. Then*

(1) $\varphi_{X+Y}(t) = \varphi_X(t)\varphi_Y(t)$
(2) $\varphi_{aX}(t) = \varphi_X(at)$

The following result relates pointwise convergence of the characteristic function and the convergence in distribution of the corresponding random variables. A proof of this theorem can be found in many probability books, for example, in Section 18.1 of Williams [**47**].

THEOREM 1.5. *(Lévy's Continuity Theorem)*
Let $\{X_n\}$ be a sequence of random variables and let $\{\varphi_n\}$ be the corresponding sequence of characteristic functions. If $\varphi_n(t) \to \varphi(t)$ for all $t \in \mathbb{R}$, then $X_n \xrightarrow{\mathcal{D}} X$.

The following important theorem is central in providing the theoretical basis for sampling from a given distribution, which we shall be doing extensively in the upcoming chapters.

THEOREM 1.6. *(Weak Law of Large Numbers)*
Let $\{X_n\}$ be a sequence of independent and identically distributed (iid) random variables with finite mean μ. Then

$$\overline{X}_n := \frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{\mathcal{P}} \mu.$$

Proof. Let $\varphi_X(t)$ be the characteristic function of the random variable X. Since μ exists, the Taylor expansion of $\varphi_X(t)$ can be expressed as follows:

$$\varphi_X(t) = 1 + it\mu + o(t), t \to 0.$$

By Proposition 1.4, we can write the characteristic function of $\overline{X}_n$ as

$$\begin{aligned}
\varphi_{\overline{X}_n}(t) &= \varphi_{\{\sum_{i=1}^n X_i\}}\left(\frac{t}{n}\right) \\
&= \left[\varphi_{X_i}\left(\frac{t}{n}\right)\right]^n \\
&= \left[1 + \frac{it\mu}{n} + o\left(\frac{t}{n}\right)\right]^n.
\end{aligned}$$

This converges pointwise to $e^{it\mu}$ as $n \to \infty$, as we shall prove in the following lemma.

LEMMA 1.7. *Let $(z_n)_{n\in\mathbb{N}}$ be a sequence of complex numbers that converges to z in $\mathbb{C}$. Then $\left(1+\frac{z_n}{n}\right)^n \to e^z$ as $n \to \infty$.*

Proof. Let $\zeta \in \mathbb{C}$ for which $|\zeta| \le 1$. Then the principal value of $\log(1+\zeta)$ has power series expansion

$$\log(1+\zeta) = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{\zeta^n}{n} = \zeta - \frac{\zeta^2}{2} + \frac{\zeta^3}{3} - \ldots$$

Thus, for any ζ for which $|\zeta| \le 1/2$,

$$|\log(1+\zeta) - \zeta| \le |\zeta|^2 \tag{1.1}$$

since

$$\begin{aligned} |\log(1+\zeta) - \zeta| &\le \frac{|\zeta|^2}{2} + \frac{|\zeta|^3}{3} + \frac{|\zeta|^4}{4} + \ldots \\ &\le \frac{|\zeta|^2}{2}(1 + |\zeta| + |\zeta|^2 + \ldots) \\ &= \frac{|\zeta|^2}{2}\frac{1}{1-|\zeta|} \\ &\le |\zeta|^2 \end{aligned}$$

Now suppose that $z_n \to z$ in $\mathbb{C}$. Since $\frac{z_n}{n} \to 0$ as $n \to \infty$, by (1.1), we have

$$\begin{aligned} n\log\left(1+\frac{z_n}{n}\right) &= n\left(\frac{z_n}{n} + o\left(\frac{1}{n^2}\right)\right) \\ &= z_n + o\left(\frac{1}{n}\right) \end{aligned}$$

which converges to z as $n \to \infty$. Therefore

$$\left(1+\frac{z_n}{n}\right)^n = \exp\left(n\log\left(1+\frac{z_n}{n}\right)\right) \to \exp(z).$$

□

Returning to the proof of the Weak Law of Large Numbers, note that since $\varphi_{\overline{X}_n}(t)$ converges to $e^{it\mu}$, by the Lévy's Continuity Theorem (Theorem 1.5), $\overline{X}_n \xrightarrow{\mathcal{D}} \mu$. Since the limit is a constant, convergence in distribution also implies convergence in probability. □

The Law of Large Numbers provides the theoretical framework for the validity of results that are drawn from an iid sample from a distribution. It can also be used to show that the histograms of these samples converge in probability to the distribution they were drawn from, as the following "histogram theorem" states.

PROPOSITION 1.8. *Let $\theta_1, \theta_2, ..., \theta_n$ be a sequence of iid random variables with probability density function π defined on $\mathbb{R}^m$. Then, for all measurable sets $A \subset \mathbb{R}^m$, one has*

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{\theta_i \in A\}} \xrightarrow{\mathcal{P}} \pi(A)$$

where $\pi(A) = \int_A \pi(\theta) d\theta$.

Proof. Recall that if $\theta_1, \theta_2, ..., \theta_n$ are independent variables, $g(\theta_1), g(\theta_2), ...,$ $g(\theta_n)$ are independent as well, provided g is a measurable function. Since A is a measurable set, $\mathbb{1}_{\{\theta_i \in A\}}$ is a measurable function of $\theta_1, \theta_2, ..., \theta_n$. Thus, letting g to be the indicator function with respect to $\theta_i \in A$, it follows that $\mathbb{1}_{\{\theta_i \in A\}}$, $i = 1, 2, ..., n$ are independent random variables. By Theorem 1.6,

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{\theta_i \in A\}} \xrightarrow{\mathcal{P}} \mathbb{E}(\mathbb{1}_{\{\theta_1 \in A\}}) = P(\theta_1 \in A) = \pi(A).$$

□

DEFINITION 1.9. *If a sample $(\theta_i)_{i=1,2,...,n}$ has the property of Proposition 1.8, then we shall say that the sample has **asymptotically the law** π.*

1.1.2. Transformations. Denote by F_X the cumulative density function (or cdf) of a continuous random variable X. We define the generalized inverse cdf of X as follows:

$$F_X^{-1}(y) = \min\{x : F_X(x) \geq y\}, y \in [0, 1].$$

With this definition, we have

$$\{F_X(X) \leq Y\} = \{X \leq F_X^{-1}(Y)\} \tag{1.2}$$

PROPOSITION 1.10. *For any continuous random variable X, the random variable $Y = F_X(X)$ has a uniform distribution over $[0, 1]$.*

Proof. Let $Y = F_X(X)$. Then clearly, the support of Y is over $[0, 1]$. Furthermore, for $y \in [0, 1]$ and using (1.2),

$$\begin{aligned} F_Y(y) &= \mathbb{P}(F_X(X) \leq y) \\ &= \mathbb{P}(X \leq F_X^{-1}(y)) \\ &= F_X(F_X^{-1}(y)) \\ &= y \end{aligned}$$

which is the cdf of a uniform $[0, 1]$ random variable. □

The following result is often used when generating random numbers from any probability distribution given its cdf by beginning from a randomly selected number between 0 and 1. For brevity, we shall denote a uniform random variable on $[a, b]$ from this point onwards as $\mathcal{U}[a, b]$.

PROPOSITION 1.11. *Let U be a $\mathcal{U}[0,1]$ random variable, and let $Y = F_X^{-1}(U)$. Then Y has the same distribution as X.*

Proof. It suffices to show that the cdf of Y is equal to F_X. Since F_X is a monotonic function, using (1.2) gives

$$F_Y(x) = \mathbb{P}(F_X^{-1}(U) \leq x) = \mathbb{P}(U \leq F_X(x)) = F_X(x),$$

as required. □

Another concept which we will need is that of orthogonal polynomials. Let S be a subset of $\mathbb{R}$ or $\mathbb{R}^m$, or, as we will have in the next chapter, a subset of the set of square integrable random variables. Then we have the following definition:

DEFINITION 1.12. *Let $\mathcal{N} = \{0, 1, ...\}$ or $\{0, 1, ..., N\}$. A system of orthogonal polynomials is a set of polynomials $\{\Phi_n\}_{n\in\mathcal{N}}$, with $\mathcal{N} \subset \mathbb{N}$ and $\deg(\Phi_n) = n$ that are orthogonal over a domain S with respect to a real positive measure α. That is, for every $m, n \in \mathcal{N}$, we have*

$$\int_S \Phi_m(x)\Phi_n(x)d\alpha(x) = \gamma_n^2\delta_{mn}, \tag{1.3}$$

where δ_{mn} is the Kronecker delta function which is 1 if $m = n$ and 0 otherwise and

$$\gamma_n^2 = \int_S \Phi_n^2(x)d\alpha(x).$$

In general, we shall assume that the measure α has a density w. In this case, (1.3) reduces to

$$\int_S \Phi_m(z)\Phi_n(z)w(z)dz = \gamma_n^2\delta_{mn},$$

if α is continuous, where the integral is replaced by a summation if α is a discrete measure. If we define the inner product of polynomials Φ_m and Φ_n as

$$\langle \Phi_m, \Phi_n \rangle := \int_S \Phi_m(z)\Phi_n(z)w(z)dz,$$

then we have the following alternative way to characterize orthogonality of Φ_m and Φ_n:

$$\langle \Phi_m, \Phi_n \rangle = \gamma_n^2\delta_{mn}$$

where $\gamma_n = \sqrt{\langle \Phi_n, \Phi_n \rangle}$.

A particular class of orthogonal polynomials is the set of *Hermite polynomials*, which are generated when α has a standard normal density. The following formula defines the standardized nth degree Hermite polynomial $H_n(x)$:

$$H_n(x) = (-1)^n e^{x^2/2} \frac{d^n}{dx^n} e^{-x^2/2}, n \in \mathbb{N}.$$

In particular, the following are the first five Hermite polynomials:

$H_0(x) = 1$ $\quad$ $H_3(x) = x^3 - 3x$

$H_1(x) = x$ $\quad$ $H_4(x) = x^4 - 6x^2 + 3$

$H_2(x) = x^2 - 1$ $\quad$ $H_5(x) = x^5 - 10x^3 + 15x$

1.1.3. The Reciprocal Gaussian Distribution. Let X be a random variable with pdf f_X, and let g be a one-to-one function of X. If g^{-1} represents the inverse of g, it is easy to show that the pdf of $Y = g(X)$ is given by $f_Y(y) = f_X(g^{-1}(y))|\frac{d}{dy}g^{-1}(y)|$ for all y in its support. Denote the normal distribution with mean μ and variance σ^2 as $N(\mu, \sigma^2)$. If we assume that $\sigma > 0$, then we can easily see from this formula that the pdf of the random variable $Y = 1/X$, where $X \sim N(\mu, \sigma^2)$ is

$$f(y) = \frac{1}{y^2\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\left(\frac{\frac{1}{y} - \mu}{\sigma}\right)^2\right\}, \quad y \neq 0 \tag{1.4}$$

REMARK 1.13. *Notice that the reciprocal Gaussian distribution is in fact a special case of a ratio distribution. In particular, it has a generalized Cauchy distribution, but where the numerator is a degenerate Gaussian distribution with mean 1 and variance 0.*

We now derive several interesting properties of this reciprocal Gaussian distribution.

First, the pdf of $Y = 1/X$ where $X \sim N(\mu, \sigma^2)$ is bimodal. Indeed, if one computes the critical points of (1.4) with respect to y, we obtain

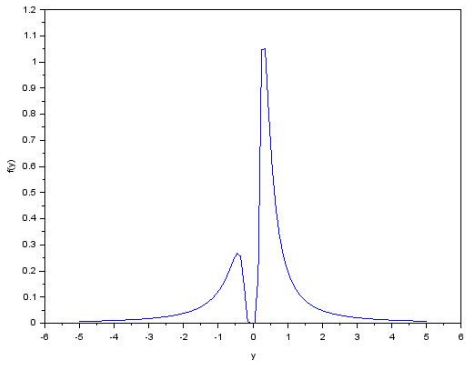
$$y = \frac{-\mu \pm \sqrt{\mu^2 + 8\sigma^2}}{4\sigma^2}. \tag{1.5}$$

This gives two distinct peaks of the distribution f of Y.

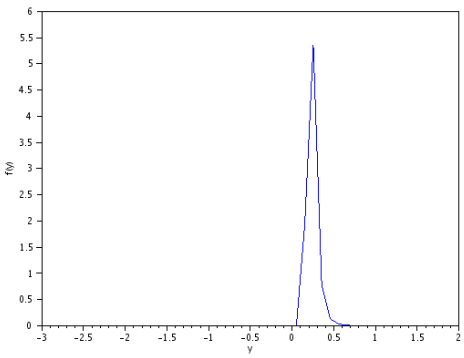
While the distribution is always bimodal, the two modes are often of different heights. In some cases, the left mode can be so small and remotely located that it is negligible with respect to the other mode. For example, in the case where $\mu = 4.5$ and $\sigma = 1$, (1.4) and (1.5) tell us that the negative critical point occurs at around $y = -2.45$, where the corresponding height is just a negligible 4×10^{-7}. Figure 1.1 shows graphs of the two cases, where the left mode is substantial in the first and insignificant in the second.

Another interesting property of this reciprocal random variable is that none of its moments exist. As we will see in the next proposition, it is linked to the fact that $f(0) > 0$ in any Gaussian distribution.

PROPOSITION 1.14. *Let X be a continuous random variable with pdf f having a support which contains 0. If $f(0) > 0$, then $\mathbb{E}(\frac{1}{X}) = +\infty$.*



(A) Two modes case. Here, $\mu = 1$ and $\sigma = 2$.



(B) "Single" mode case. Here, $\mu = 4.5$ and $\sigma = 1$. The second mode at $y \approx -2.45$ is virtually invisible.

FIGURE 1.1. Two possible graphs of the pdf of $Y = 1/X$, where X is normally distributed.

Proof. By definition, and decomposing $x \mapsto f(x)/x$ into its positive and negative parts,

$$\begin{aligned} \mathbb{E}\left(\frac{1}{X}\right) &= \int_{-\infty}^{\infty} \frac{1}{x} f(x)dx \\ &= \int_{0}^{\infty} \frac{1}{x} f(x)dx - \int_{-\infty}^{0} \frac{1}{|x|} f(x)dx \end{aligned} \tag{1.6}$$

We show that the first term of (1.6) is infinite (both are, actually), and thus the expression is undefined. Since f is continuous at 0, then for every $\epsilon > 0$, we can find a $\delta > 0$ such that $|x| < \delta$ implies $|f(x) - f(0)| < \epsilon$. Now suppose we choose $\epsilon = f(0) - b$, where $0 < b < f(0)$. Then, there exists a $\delta > 0$ such that $f(x) > b$ for all $x \in (-\delta, \delta)$. Thus,

$$\begin{aligned} \int_{0}^{+\infty} \frac{1}{x} f(x)dx &\geq \int_{0}^{\delta} \frac{1}{x} f(x)dx \\ &\geq \int_{0}^{\delta} \frac{b}{x} dx \\ &= +\infty \end{aligned}$$

□

PROPOSITION 1.15. *Let r and s be positive integers with $s > r$. If the rth moment of a random variable X is not finite, then the sth moment of X is not finite as well.*

Proof. We prove the contrapositive of this statement instead. That is, we show that if the sth moment is finite, then the rth moment must be finite as well. Consider the function $f(x) = x^{s/r}$. Since $s > r$, f is convex. By Jensen's

inequality,

$$\mathbb{E}(|X|^s) = \mathbb{E}[(|X|^r)^{s/r}] \geq \{\mathbb{E}(|X|^r)\}^{s/r}.$$

If $E(|X|^r) < 1$, then $E(|X|^r)$ is obviously finite. On the other hand, if $E(|X|^r) \geq 1$, then $\{\mathbb{E}(|X|^r)\}^{s/r} \geq \mathbb{E}(|X|^r)$, and so the rth moment must be finite because $\mathbb{E}(|X|^r)$ is bounded above by the finite value $\mathbb{E}(|X|^s)$. Therefore, in either scenario, we have proven that the rth moment is finite. □

Combining the two previous propositions, it is then clear that $Y = \frac{1}{X}$, where $X \sim N(\mu, \sigma^2)$, has no finite moments.

REMARK 1.16. *The existence of the moments of a random variable is central to many major results in classical probability theory. For example, the Central Limit Theorem and Chebyshev's Inequality require the first two moments of a random variable to exist. The Law of Large Numbers, on the other hand, assumes that the mean is well-defined. In the situation where the moments exist, studying the properties of a random variable can be fairly straightforward, as most of the classical results are at our disposal.*

1.1.4. Markov chains. Since our state space is typically $\mathbb{R}^m$ or a subset of it, we need to consider general state space Markov chain theory. To facilitate understanding, we have chosen to present all definitions and results using $\mathbb{R}^m$ as the state space instead of a more general space E. The following exposition shall be mainly based on the book by Robert and Casella [**34**].

DEFINITION 1.17. *Let $\mathcal{B}(\mathbb{R}^m)$ be the set of Borel subsets of $\mathbb{R}^m$. The **transition kernel** is a function K defined on $\mathbb{R}^m \times \mathcal{B}(\mathbb{R}^m)$ such that*

(i) $\forall \theta \in \mathbb{R}^m$, $K(\theta, \cdot)$ *is a probability measure*
(ii) $\forall B \in \mathcal{B}(\mathbb{R}^m)$, $K(\cdot, B)$ *is measurable.*

DEFINITION 1.18. *Given a transition kernel K, a sequence $X_0, X_1, ..., X_n, ...$ of random variables is a **Markov chain** of kernel K, denoted by $(X_n)_{n\in\mathbb{N}}$ if, for any n and any $B \in \mathcal{B}(\mathbb{R}^m)$,*

$$\mathbb{P}(X_{n+1} \in B | X_0, ..., X_n) = \mathbb{P}(X_{n+1} \in B | X_n) \tag{1.7}$$

$$= \int_B K(X_n, dx). \tag{1.8}$$

The following lemma, which will be useful to prove that the sequence we will construct is a Markov chain, is an extension to non-countable sets of a result by Pardoux [**31**].

LEMMA 1.19. *Let h be a mapping from $\mathbb{R}^m \times \mathbb{R}^m$ into $\mathbb{R}^m$. Let $X_0, Y_1, Y_2, ...$ be mutually independent $\mathbb{R}^m$-valued random variables and $(X_n)_{\{n\in\mathbb{N}\}}$ be defined*

recursively by

$$X_{n+1} = h(X_n, Y_{n+1}), n \in \mathbb{N}.$$

Then $\{X_n; n \in \mathbb{N}\}$ *is a Markov chain.*

Proof. First, we claim that for any $i = 1, 2, ..., n$, X_i can be written as a function of $X_0, Y_1, Y_2, ...Y_i$ alone. If this is true, then Y_{n+1} will be independent of $X_0, X_1, ..., X_n$ by the Disjoint Blocks Theorem (see Theorem 3.10, page 76 of [**18**]) since $X_0, X_1, ..., X_n$ would then be functions of random variables different from and all independent of Y_{n+1}.

To prove our claim, we use a simple induction argument. Certainly, the claim is true when $i = 2$ as $X_2 = h(X_1, Y_2) = h(h(X_0, Y_1), Y_2)$, which is exclusively a function of X_0, Y_1, Y_2. Now suppose that X_k can be written as a function of $X_0, Y_1, ..., Y_k$. By definition, $X_{k+1} = h(X_k, Y_{k+1})$. But X_k is a function of $X_0, Y_1, ..., Y_k$ by the inductive hypothesis, and so X_{k+1} is a function of $X_0, Y_1, ..., Y_{k+1}$.

Now denote by $f_{X_0, X_1, ..., X_k}$ the joint distribution of $X_0, X_1, ..., X_k$. Then

$$\begin{aligned}
\mathbb{P}(X_{n+1} \in B | X_0, X_1, ..., X_n) &= \int_B \frac{\mathbb{P}(X_0 \in dx_0, ..., X_{n+1} \in dx_{n+1})}{\mathbb{P}(X_0 \in dx_0, ..., X_n \in dx_n)} \\
&= \int_{h(X_n, Y_{n+1}) \in B} \frac{\mathbb{P}(X_0 \in dx_0, ..., X_n \in dx_n, Y_{n+1} \in dy_{n+1})}{\mathbb{P}(X_0 \in dx_0, ..., X_n \in dx_n)} \\
&= \int_{h(X_n, Y_{n+1}) \in B} \mathbb{P}(Y_{n+1} \in dy_{n+1}) \\
&= \int_B \mathbb{P}(X_{n+1} \in dx_{n+1} | X_n) \\
&= \mathbb{P}(X_{n+1} \in B | X_n)
\end{aligned}$$

where we used the claim to obtain the third equality from the second. □

We now give a quick summary of the important properties and results in Markov chain theory that we will need later.

Definition 1.20. *A Markov chain is* φ*-irreducible for a probability measure* φ *on* $\mathbb{R}^m$ *if for all measurable sets* $A \subset \mathbb{R}^m$ *with* $\varphi(A) > 0$*, we have*

$$\mathbb{P}(\tau_A < \infty | X_0 = x) > 0 \quad \forall x \in \mathbb{R}^m.$$

where

$$\tau_A := \inf\{n \in \mathbb{N} : X_n \in A\}.$$

A Markov chain is irreducible if it is φ*-irreducible for some probability distribution* φ*.*

In simple terms, irreducibility means that all "interesting" sets of $\mathbb{R}^m$ can be reached, regardless of the starting point x.

DEFINITION 1.21. *(Def. 6.19 in* [**34**]*) A φ-irreducible chain (X_n) is small if there exists an $m \in \mathbb{N}^*$ and a nonzero measure $\nu_m(A)$ such that*

$$K^m(x, A) \geq \nu_m(A)$$

for all $x \in C$ and all $A \in \mathcal{B}(\mathbb{R}^m)$.

DEFINITION 1.22. *(Def. 6.23 in* [**34**]*) A φ-irreducible chain (X_n) has a cycle of length d if there exists a small set C, an associated integer M, and a probability distribution ν_M such that d is the gcd of the set*

$$\{m \geq 1; \exists \delta_m > 0 \text{ such that } C \text{ is small for } \nu_m \geq \delta_m \nu_M\}.$$

It can be shown that the number d is independent of the small set C and thus intrinsically characterizes a Markov chain (X_n). If the largest integer d satisfying Definition 1.22 is 1, then we say that (X_n) is *aperiodic*.

DEFINITION 1.23. *A probability measure π is said to be* invariant *for the transition kernel K (and for the associated chain) if*

$$\pi(B) = \int_S K(x, B)\pi(x)dx, \quad \forall B \in \mathcal{B}(\mathbb{R}^m)$$

where S is the state space of the Markov chain.

In this case, the invariant distribution is also referred to as stationary since $X_0 \sim \pi$ implies that $X_n \sim \pi$ for every n, which means that the Markov chain is stationary in its distribution.

An alternative way to prove that a certain distribution $\pi(x)$ is the stationary distribution of a Markov chain is to show that the Markov chain satisfies the *detailed balance* property, which is made precise in the following lemma.

LEMMA 1.24. *Suppose that a Markov chain with transition kernel K satisfies $\pi(a)K(a, b) = \pi(b)K(b, a)$ for some probability distribution $\pi(a)$. Then $\pi(a)$ is the stationary distribution of the chain.*

Proof. Let S be the state space of the Markov chain. For any measurable set B,

$$\begin{aligned}
\int_{a\in S} K(a, B)\pi(a)da &= \int_{a\in S}\int_{b\in B} K(a, b)\pi(a)db\, da \\
&= \int_{a\in S}\int_{b\in B} K(b, a)\pi(b)db\, da \\
&= \int_{b\in B}\left(\int_{a\in S} K(b, a)da\right)\pi(b)db \\
&= \int_{b\in B} \pi(b)db,
\end{aligned}$$

since $\int_{a\in S} K(b,a)da = 1$. □

Before we can consider what happens with the long-term behavior of a Markov chain, it is necessary to define what metric we will use to compare the distributions. If μ and ν are two measures defined on $\mathbb{R}^m$, we shall consider the *total variation distance* norm:

$$\|\mu - \nu\| := \sup_{A\subset E} |\mu(A) - \nu(A)| \tag{1.9}$$

where A must be measurable. Suppose we denote the nth transition probabilities by $K^n(x,\cdot)$. More precisely, $K^1(x,A) := K(x,A)$ and $K^{n+1}(x,A) := \int_{\mathbb{R}^m} K^n(x,dy)K(y,A)$ for $n \in \mathbb{N}$. Then, we have the following asymptotic result.

THEOREM 1.25. *Suppose $(X_n)_{n\geq 0}$ is an irreducible, aperiodic Markov chain on $\mathbb{R}^m$ with transition kernel K and stationary distribution π. Then*

$$\|K^n(x,\cdot) - \pi(\cdot)\| \to 0$$

under the total variation norm (1.9) *for π-a.e. and for all $x \in \mathbb{R}^m$.*

The proof of this theorem is based on several lemmas which describe properties of Markov chains with respect to irreducibility and aperiodicity. One may refer to Meyn and Tweedie [**27**] or Casella and Robert [**34**] for the detailed proof.

1.2. Differential Equations

In this section, we give several results based on an l-dimensional differential system

$$\frac{dy}{dt} = g(t,y), y(t_0) = y_0. \tag{1.10}$$

where $y : \mathbb{R} \to \mathbb{R}^l$, $g : \mathbb{R} \times \mathbb{R}^l \to \mathbb{R}^l$, and $y_0 \in \mathbb{R}^l$.

In later sections, we shall occasionally need to look at second-order Taylor expansions of functions of the coefficients of a system of differential equations. For such an expansion to be well-defined, it is necessary that the differential system satisfies certain properties. For this, we need to review a few theorems in differential systems theory.

For the first theorem, we shall express the solution $y(t)$ and the initial point y_0 in terms of its components as follows:

$$y(t;y_0) = \{y_1(t,y_0), y_2(t,y_0), ..., y_l(t,y_0)\}$$

$$y_0 = \{y_{10}, y_{20}, ..., y_{l0}\}$$

THEOREM 1.26. *Let $g(t,y)$ be continuous and satisfy a Lipschitz condition on y on the region R defined by*

$$\|y - y_0\| \leq a, \quad |t - t_0| \leq b.$$

Then there exist $a' > 0$ and $b' > 0$ such that the solution $y(t, y_0)$ of (1.10), considered as a function of t and its initial value y_0 is of class C^1 with respect to both Y_0 and t simultaneously, for any (t, y_0) in a region

$$\|y - y_0\| \leq a' < a, \quad |t - t_0| \leq b' < b.$$

The proof of this theorem can be found in most differential equation textbooks, for example, as Theorems 8 and 9 in Hurewicz [**16**].

This theorem is a local result that, in fact, is still true more globally. When the solution $y(t, y_0)$ exists in a region, then it is a C^1 function of (t, y_0) simultaneously in the whole region as C^1 is a local property.

COROLLARY 1.27. *Consider a system of differential equations in which the functions g_i depend upon any number of coefficients $\mu_1, ..., \mu_m$*

$$\frac{dy_i}{dt} = g_i(y_1, y_2, ..., y_l; \mu_1, \mu_2, ..., \mu_m; t), \quad i = 1, 2, ..., l \tag{1.11}$$

If each of the g_i's has partial derivatives with respect to $y_1, y_2, ..., y_l; \mu_1, \mu_2, ..., \mu_m$ continuous in some $(l + m + 1)$-dimensional region R, then the solutions

$$y_i(t; y_{10}, y_{20}, ..., y_{l0}; \mu_1, \mu_2, ..., \mu_m), \quad i = 1, 2, ..., l \tag{1.12}$$

will have partial derivatives in $\mu_1, \mu_2, ..., \mu_m$ continuous in all their arguments over any subset of R where the solutions (1.12) *are defined.*

Proof. We consider the coefficients $\mu_1, \mu_2, ..., \mu_m$ as new variables, and add the equations

$$\frac{d\mu_j}{dt} = 0, \quad j = 1, 2, ..., m \tag{1.13}$$

to the system (1.11). Then, all the conditions of Theorem 1.26 are still satisfied by the system formed by combining the equations (1.11) and (1.13). Hence, the solutions y_i are of class C^1 with respect to "initial values" of the μ_j's. But since the μ_j's are taken as constants with respect to t, the result follows. □

In general, we shall require that our solutions are twice differentiable with respect to their coefficients. It turns out that by the two previous results, we simply need that f be also of class C^2 as well.

COROLLARY 1.28. *If f is of class C^2, then the solution $y(t, y_0; \lambda)$ of the differential system $\frac{dy}{dt} = g(y, \lambda)$ is also of class C^2.*

Proof. Our proof will be based on that given by Cartan in [**2**]. It suffices to prove that the partial derivatives $\varphi'_y(y, \lambda)$ and $\varphi'_\lambda(y, \lambda)$ of the solution φ with respect to y and the coefficients λ are of class C^1.

To do this, we first note that $\varphi'_y(y, \lambda)$ is the solution of the differential system

$$\frac{dy}{dt} = g'_y(\varphi(u, \lambda), \lambda) \cdot y(t), \quad y(t_0) = 1,$$

where the right-hand side of the equation is of class C^1 in both x and λ. Thus, its solution must be a function of class C^1 by Theorem 1.26. Similarly, $\varphi'_\lambda(x,\lambda)$ is the solution of

$$\frac{dz}{dt} = f'_x(\varphi(u,\lambda),\lambda)\cdot z(t) + f'_\lambda(\varphi(t,u,\lambda),\lambda)$$

where $z(t_0) = 0$. Once again, the right-hand side of the equation is of class C^1, thus the solution is also of class C^1 in (u,λ) by Theorem 1.26. Thus $\varphi'_x(x,\lambda)$ and $\varphi'_\lambda(x,\lambda)$ are both of class C^1, as required. □

1.3. Higher-dimensional ellipsoids

PROPOSITION 1.29. *Let A be a positive definite $l\times l$ matrix and $x\in\mathbb{R}^l$. The volume of the l-dimensional ellipsoid $A_\epsilon = \{x\in\mathbb{R}^l, x'Mx<\epsilon\}$ is given by*

$$V(A_\epsilon) = \frac{(\pi\epsilon)^{l/2}}{\Gamma(\frac{l}{2}+1)}\cdot(\det M)^{-1/2}.$$

where Γ is the gamma function defined by

$$\Gamma(u) = \int_0^\infty z^{u-1}e^{-z}dz.$$

Proof. It suffices to compute for the volume V_l of the l-dimensional sphere of radius $\sqrt{\epsilon}$. This is because the l-dimensional ellipsoid $x^TAx<\epsilon$ is just a linear transformation of the l-dimensional sphere of radius ϵ using the transformation $x = A^{-1/2}y$, so we can then deduce that

$$V(A_\epsilon) = (\det A)^{-1/2}\cdot V_l. \tag{1.14}$$

To compute the volume of the l-dimensional sphere of radius r, we compute its surface area S_l indirectly. To do this, we first note that

$$\left(\int_{-\infty}^{\infty} e^{-x^2}dx\right)^l = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} e^{-(x_1^2+x_2^2+\ldots+x_l^2)}dx_l dx_{l-1}\ldots dx_1. \tag{1.15}$$

Converting to hyperspherical coordinates, we have

$$x_1^2+x_2^2+\ldots+x_l^2 = r^2$$

and

$$dx_1dx_2\ldots dx_l = r^{l-1}drd\Omega_{l-1}.$$

where $d\Omega_{l-1}$ contains all the angular factors. Thus,

$$\begin{aligned}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} e^{-(x_1^2+x_2^2+\ldots+x_l^2)}dx_1\ldots dx_l &= \int_0^{\infty} e^{-r^2}r^{l-1}dr\underbrace{\int d\Omega_{l-1}}_{S_l}\\ &= \int_0^{\infty} e^{-r^2}r^{l-2}\cdot\frac{1}{2}\cdot 2rdr\cdot S_l\\ &= \int_0^{\infty} e^{-r^2}(r^2)^{\frac{l}{2}-1}\cdot\frac{1}{2}\cdot 2rdr\cdot S_l\\ &= \frac{1}{2}\int_0^{\infty} e^{-u}u^{\frac{l}{2}-1}du\cdot S_l\\ &= \frac{1}{2}\Gamma\left(\frac{l}{2}\right)\cdot S_l\end{aligned}$$

Since $\int_{-\infty}^{+\infty} e^{-x^2}dx = \sqrt{\pi}$, the left side of (1.15) has a value of $\pi^{l/2}$. Thus, we have

$$\pi^{l/2} = \frac{1}{2}\Gamma\left(\frac{l}{2}\right)S_l$$

which gives

$$S_l = \frac{\pi^{l/2}}{\frac{1}{2}\Gamma(\frac{l}{2})} \tag{1.16}$$

The volume of our l-dimensional sphere is then

$$V_l(\sqrt{\epsilon}) = \int_0^{\sqrt{\epsilon}} S_l\cdot r^{l-1}dr = \frac{S_l\epsilon^{l/2}}{l}.$$

Substituting in (1.16) yields

$$V_l = \frac{(\pi\epsilon)^{l/2}}{\frac{l}{2}\Gamma(\frac{l}{2})} = \frac{(\pi\epsilon)^{l/2}}{\Gamma(\frac{l}{2}+1)} \tag{1.17}$$

Substituting back to (1.14),

$$V(A_\epsilon) = \frac{(\pi\epsilon)^{l/2}}{\Gamma(\frac{l}{2}+1)}\cdot(\det A)^{-1/2} \tag{1.18}$$

□

PROPOSITION 1.30. *Let $X = (X_1, X_2, ..., X_m)$ be the random vector that represents the coordinates of a point chosen uniformly on or within the unit m-sphere $S_0 = \{x \in \mathbb{R}^m : ||x|| \le 1\}$. Then $\mathbb{E}(X_i^2) = \frac{1}{m+2}$ for $i = 1, 2, ..., m$.*

The idea of the proof is to note that for all $x \in \mathbb{R}^m$, the limit of

$$\frac{\mathbb{P}(r \le ||x|| \le r + dr)}{r^{m-1}dr}$$

is a constant C as $dr \to 0$. Since $\int_0^1 C\cdot r^{m-1}dr = 1$, then $C = m$. Thus,

$$\mathbb{E}(||x||^2) = \int_0^1 mr^{m+1}dr = \frac{m}{m+2}. \tag{1.19}$$

By the symmetry of the support, $\mathbb{E}(\|x\|^2) = m\mathbb{E}(X_i^2)$ for all $i \in \{1, 2, ..., m\}$. Combining this and (1.19) gives the desired result.

1.4. Five Main Examples

In this section, we introduce the five different systems of differential equations that we will use to test the methods that will be introduced in the next chapters.

1.4.1. Logistic Model. The logistic model is one of the simplest non-linear differential equations. It is used to model the growth of a quantity y which exhibits a damping effect. That is, it grows more slowly as it approaches a certain threshold value K. It is given by the following differential equation:

$$\frac{dy}{dt} = ry\left(1 - \frac{y}{K}\right)$$

where r represents the intrinsic growth rate and K the threshold. This model is used to represent various phenomena including the growth of a population, the concentration of reactants and products, or even the saturation of a market.

In most practical applications, the initial point y_0 of a logistic model is always in between its two constant solutions $y = 0$ and $y = K$. However, if we allow y_0 to be negative or to be above the threshold value K, we can encounter the problem of explosion in finite time. This means that there exists a finite time $\bar{t}$ for which $\lim_{t\to\bar{t}} y(t)$ is equal to $+\infty$ or $-\infty$. We shall see later in Section 2.2 that this can become an obstacle for the simulation of the solution at a fixed time t^* of a differential equation with random coefficients.

1.4.2. Harmonic Oscillator. The harmonic oscillator has its roots in classical mechanics in physics. It is used to represent any system that experiences a restoring force F proportional to the displacement, x. For example, in a spring-mass system, we know that when a spring is stretched or compressed by a mass for a certain length, the spring exerts a force proportional to (but to the opposite direction of) the displacement. We shall only look at the case of a *simple* harmonic oscillator, where the only force acting on the system is F. By using Newton's second law, we can write this system in terms of the second-order differential equation as

$$\frac{d^2x}{dt^2} = -kx$$

where k is an elastic coefficient. This can be written as the following system of differential equations:

$$\begin{cases} \dfrac{dx}{dt} = -ay \\ \dfrac{dy}{dt} = bx \end{cases} \tag{1.20}$$

where a and b are unknown positive coefficients.

It is not difficult to show that the general solution to this system is given by

$$\begin{cases} x(t) &= K_1 \cos\sqrt{ab}t - K_2 \sin\sqrt{ab}t \\ y(t) &= K_1\sqrt{\frac{b}{a}} \sin\sqrt{ab}t + K_2\sqrt{\frac{b}{a}} \cos\sqrt{ab}t \end{cases} \tag{1.21}$$

where K_1 and K_2 are arbitrary constants.

1.4.3. Lotka-Volterra Model. The Lotka-Volterra Model is a classic model of predator-prey population dynamics. If we let x and y represent the size of the population of a prey and a predator, respectively, we can write the model as a system of differential equations with four coefficients α, β, γ and δ as follows:

$$\begin{cases} \frac{dx}{dt} &= \alpha x - \beta xy \\ \frac{dy}{dt} &= \gamma xy - \delta y \end{cases} \tag{1.22}$$

The coefficients α and γ represent the growth rates of the prey and predator, respectively. On the other hand, β is the rate of predation and δ is the loss rate of predators through means such as natural death or emigration. In the model, the predators thrive when there are plentiful prey. However, once the predator population outstrips the prey population, the predators decline in number. This allows the prey population to increase again, and this cycle of growth and decline continues periodically. Like the harmonic oscillator, as the dynamics of this system are well-understood and somewhat regular, it will be used initially to study the basic properties of our coefficient estimation methods.

It is possible to reduce this system to one involving only two unknown coefficients. This can be done by making the substitution $X = x$, $Y = \frac{\beta}{\delta}y$, and $\tau = \delta t$. In this case,

$$\begin{aligned} \frac{dX}{d\tau} &= \frac{dX}{dt} \cdot \frac{dt}{d\tau} \\ &= \frac{1}{\delta} \cdot \frac{dX}{dt} \\ &= \frac{1}{\delta}\left(\alpha X - \beta X \cdot \frac{\delta}{\beta} Y\right) \\ &= AX - XY \end{aligned}$$

if we let $A = \frac{\alpha}{\delta}$. Similarly, we can write

$$\begin{aligned}\frac{dY}{d\tau} &= \frac{dY}{dt}\cdot\frac{dt}{d\tau}\\ &= \frac{1}{\delta}\cdot\frac{dY}{dt}\\ &= \frac{1}{\delta}\cdot\frac{\beta}{\delta}\cdot\left(\gamma X\cdot\frac{\delta}{\beta}Y-\delta\cdot\frac{\delta}{\beta}Y\right)\\ &= BXY-Y\end{aligned}$$

by choosing $B = \frac{\gamma}{\delta}$.

1.4.4. The Repressilator. The repressilator is a popular toy model for gene regulatory systems which was proposed by Michael B. Elowitz and Stanislas Leibler in 2000 [**9**]. It consists of three genes connected in a feedback loop, where each gene transcribes the repressor protein for the next gene in the loop. The dynamic of the messenger RNAs of the three geners are given by $m_1(t)$, $m_2(t)$, $m_3(t)$, while the dynamic of the three repressor-proteins produced are represented by $p_1(t)$, $p_2(t)$, and $p_3(t)$. Transcription and degradation are assumed to have a linear dynamic while repression is given by a nonlinear term $\alpha/(1+p^n)$. The model is represented by the following system of six equations and four coefficients $(\alpha_0, \gamma, \alpha, \beta)$:

$$\left\{\begin{array}{rcl}\frac{dm_1}{dt} &=& -m_1+\frac{\alpha}{1+p_3^\gamma}+\alpha_0\\ \frac{dp_1}{dt} &=& -\beta(p_1-m_1)\\ \frac{dm_2}{dt} &=& -m_2+\frac{\alpha}{1+p_1^\gamma}+\alpha_0\\ \frac{dp_2}{dt} &=& -\beta(p_2-m_2)\\ \frac{dm_3}{dt} &=& -m_3+\frac{\alpha}{1+p_2^\gamma}+\alpha_0\\ \frac{dp_3}{dt} &=& -\beta(p_3-m_3)\end{array}\right. \tag{1.23}$$

For most of the values of coefficients, the dynamic is oscillatory and shows sustained oscillations for some specific values of the coefficients.

1.4.5. A Simplified Circadian Cycle Model. The most complex model that we will be using is that of a simplified model for the mammalian circadian clock constructed by Comet et. al. in [**3**]. It is based on a model proposed by Leloup and Golbeter for circadian oscillations in mammals involving interlocked negative and positive regulations of certain genes by their protein products and consisting of 16 differential equations. By excluding the dynamics of one protein

and the phosphorylation of several other proteins, Comet successfully transformed this system to one that still adequately captures the 24-hour oscillatory behavior of the original model, but now consisting of only 4 equations and 12 coefficients $(K, \gamma, k_1, k_2, k_3, k_4, kd_1, kd_2, kd_3, kd_4, v_1, v_2)$. The model is given by the following system:

$$\left\{\begin{array}{rcl} \dfrac{dp_1}{dt} & = & v_1 \dfrac{K^\gamma}{K^\gamma + c_2^\gamma} - k_3 p_1 p_2 + k_4 c_1 - kd_1 p_1 \\ \dfrac{dp_2}{dt} & = & v_2 \dfrac{K^\gamma}{K^\gamma + c_2^\gamma} - k_3 p_1 p_2 + k_4 c_1 - kd_2 p_2 \\ \dfrac{dc_1}{dt} & = & k_3 p_1 p_2 - k_4 c_1 - k_1 c_1 + k_2 c_2 - kd_3 c_1 \\ \dfrac{dc_2}{dt} & = & k_1 c_1 - k_2 c_2 - kd_4 c_2 \end{array}\right. \tag{1.24}$$

where the equations represent the dynamics of cystolic PER protein (p_1), CRY protein (p_2), cystolic PER-CRY protein complex (c_1) and nuclear PER-CRY protein complex (c_2). We will use this 4-dimensional differential system to test our methods on a system with a large number of coefficients.

CHAPTER 2

Law of the Solution at time t^* of a Differential Equation with Random Coefficients

In this chapter, we wish to study the law of the solution at time t^* of a differential equation with random coefficients. In Section 2.1, we begin to examine the problem in what is probably the simplest case, which is on a linear differential equation. We shall show that even in this case, our only option to have an idea on the distribution of the solution may be Monte Carlo simulation. In the next section (2.2), we give an example of where direct Monte Carlo simulation may not even be possible. As an alternative, in the final section, we give a short exposition of *polynomial chaos*, which may be used to approximate certain random variables using orthogonal polynomials as basis. We illustrate how this method can be used to study the law of the solution at time t^* of certain differential systems.

2.1. An example in the linear case

Consider a differential equation of the form

$$y' = g(y;\theta)$$

with one or more unknown coefficients $\theta = (\theta^1, \theta^2, ..., \theta^m) \in \mathbb{R}^m$ and a fixed initial condition $y(0) = y_0$. We assume that the coefficient θ is a random variable and that we know its law. Then, for a fixed time $t^* > 0$, $y(t^*)$ is a random variable which depends on θ. We are interested in the general problem of studying the probability distribution function (pdf), or law, of $y(t^*)$. When the system possesses an explicit solution, the question seems to be easy. We shall see shortly that this problem, even in the easiest cases, is not as simple as it seems.

As a first example, consider the very simple linear differential equation $y' = -Ay + B$, where the initial point $y(0) = y_0$ is fixed, and exactly one of A and B is a random variable. Assume first that $A = a$ is fixed, and $B \sim N(\mu, \sigma^2)$. To determine the law of $y(t^*)$, we begin by computing the solution to the linear differential equation. The explicit solution of the differential equation at time t^* is given by

$$y(t^*) = \left(y_0 - \frac{B}{a}\right) e^{-at^*} + \frac{B}{a} \tag{2.1}$$

We can rewrite this as

$$y(t^*) = y_0 e^{-at^*} + B\left(\frac{1 - e^{-at^*}}{a}\right),$$

which we can identify as a linear function of the Gaussian random variable B. Thus, $y(t^*) \sim N(\widetilde{\mu}, \widetilde{\sigma}^2)$, where

$$\widetilde{\mu} = \mu\left(1 - \frac{e^{-at^*}}{a}\right) + y_0 e^{-at^*}$$

and

$$\widetilde{\sigma}^2 = \sigma^2\left(1 - \frac{e^{-at^*}}{a}\right)^2.$$

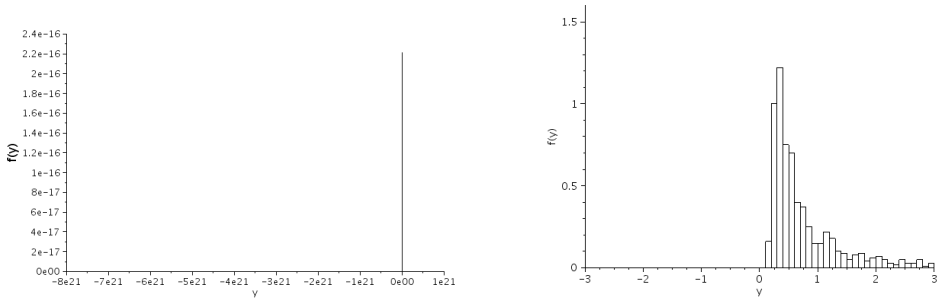
Thus, we obtain a family of Gaussian distributions depending on the chosen values of y_0, a and t^*. Furthermore, when t^* becomes large, the family of Gaussian laws $N(\widetilde{\mu}, \widetilde{\sigma}^2)$ has a different limiting behaviour depending on the sign of a. If $a > 0$, the first term of (2.1) tends to 0 regardless of the value of B. Therefore, we would expect the law of $y(t^*)$ to be close to the Gaussian distribution with mean μ/a and variance σ^2/a^2. On the other hand, if $a < 0$, then $y(t^*)$ does not tend to any distribution. This is because $y(t^*)$ converges to either $+\infty$ or $-\infty$, depending on the sign of $y_0 - \frac{B}{a}$. As $\lim_{t^* \to \infty} \sigma^2 = +\infty$, we would therefore expect $y(t^*)$ to consist of arbitrarily spread large positive and large negative values.

When A is random while $B = b$ is fixed, the situation becomes a lot more complicated. In this case, the solution is now

$$y(t^*) = \left(y_0 - \frac{b}{A}\right) e^{-At^*} + \frac{b}{A} \tag{2.2}$$

In this case, the second term of (2.2) has a form similar to that of the reciprocal Gaussian distribution in Section 1.1.3. On the other hand, the first term is a product of two random variables, with the first looking like a shifted reciprocal Gaussian, while the second being a lognormal random variable. The resulting distribution is certainly not one of the well-known distributions. Furthermore, we cannot write analytically its distribution.

The most natural way to obtain an idea of the shape of this distribution is through a typical Monte Carlo simulation. Suppose we fix b to be equal to 1, and consider the differential equation $y' = -Ay + 1$, where $A \sim N(1, 4)$, the initial point $y_0 = -1$, and $t^* = 10$. To obtain an estimate of the pdf, we generate 1000 values of A using `grand` function of Scilab, and compute the solution of the differential equation at time t^* using the `ode` command. The `histplot` command can then be used to construct the histogram of the resulting values $y(t^*)$. Figure 2.1a shows the histogram of the resulting values if we let Scilab automatically choose the classes of the histogram.



(A) Histogram after letting scilab choose the classes

(B) Histogram over the interval $(-3, 3)$

FIGURE 2.1. Histograms of the values of the solution of the differential equation $y' = -Ay + 1$ when $t^* = 10$ and A takes of 1000 values from $N(1, 4)$. The histograms were drawn using the `histplot` command in Scilab and 50 classes.

Clearly, this histogram does not provide a satisfactory result. This is because Scilab automatically constructs classes with equal class sizes between the lowest and highest values $y(t^*)$. However, in this case, the range of values for $y(t^*)$ is too wide, and the frequency of larger values seems to be small compared to the smaller values. Thus, all the smaller values were bunched up into a single class.

Since most of the sampled values of $y(t^*)$ seem to be small values, one possible workaround is to draw the histogram only over a small interval around 0. Figure 2.1b shows the histogram over $(-3, 3)$. While this seems to solve the problem of displaying the histogram, estimating the area under the curve for the given interval shows that the area is clearly less than 1. Thus, our graph fails to account for a good portion of the observations. Recalling the graph of the reciprocal Gaussian distribution from the previous chapter (see Figure 2.2 below), we see that it seems to be missing one of its modes.

A third, and better, option would be to "cut" the values at some point, which we now describe more precisely. Let $a_1, a_2, ..., a_n$ be a sample from A, and let $\{y_i\} \coloneqq \{y(t^*; y_0, a_i)\}$ be the values of the corresponding solutions at time t^*. Choose an upper bound y_{hi} and a lower bound y_{lo}. Then for each i where $y_i > y_{hi}$, re-define $y_i = y_{hi}$. Otherwise, y_i remains the same. A similar procedure can be done for those values which are below y_{lo}. Since all the values are now in between -5 and 5, we can display the whole histogram properly, as shown in Figure 2.2, but also considering the extreme values.

Figure 2.2 helps us understand why we were having a lot of difficulty in displaying our histogram properly. Although they do not seem to be much when taken as a class of the original histogram, there are in fact a lot of values with high absolute value on both ends, as indicated by the two outer bars.

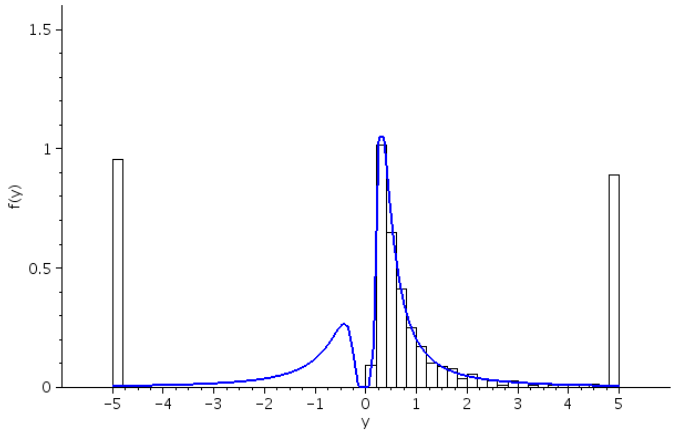


FIGURE 2.2. Histogram of $\{y_i\}$. The theoretical pdf of the reciprocal Gaussian $1/X$, where $X \sim N(1,4)$ is superimposed to the graph.

Can we approximate the percentage of the distribution represented by each of the two peaks in the side? The answer is yes, and it turns out that this follows rather easily from the solution of the linear differential equation. Going back to (2.2), we see that for $A > 0$, $y(t^*)$ is close to $\frac{1}{A}$ when t^* is large enough. On the other hand, if $A < 0$, $y(t^*)$ behaves like $(y_0 - \frac{1}{A})e^{-At^*}$ when t^* is large enough, which is either a large positive or a large negative value depending on the sign of $y_0 - \frac{1}{A}$.

The above observations can be summarized in the following result, which allows us to approximate the percentage of observations represented by the two bars at the ends.

PROPOSITION 2.1. *Consider the differential equation* $y' = -Ay + 1$, *where* $y(0) = y_0$. *Then*

$$(1)\ \left\{A|\lim_{t\to\infty} y(t) = -\infty\right\} = \begin{cases} \varnothing, & \text{if } y_0 > 0 \\ \left\{A|A < \frac{1}{y_0}\right\}, & \text{otherwise} \end{cases}$$

$$(2)\ \left\{A|\lim_{t\to\infty} y(t) = \infty\right\} = \begin{cases} \{A|A < 0\}, & \text{if } y_0 > 0 \\ \left\{A|\frac{1}{y_0} < A < 0\right\}, & \text{otherwise} \end{cases}$$

Proof. To show (1), we recall from (2.2) that the solution to the linear differential equation can be written analytically as

$$y(t) = \left(y_0 - \frac{1}{A}\right)e^{-At} + \frac{1}{A}.$$

As $t \to \infty$, $y(t) \to -\infty$ if and only if $A < 0$ and $y_0 < \frac{1}{A}$. If y_0 is positive, then no such A exists since $1/A$ is always negative. On the other hand, if $y_0 < 0$, then

$$\begin{aligned}\left\{A|A<0\cap y_0<\frac{1}{A}\right\} &= \{A|Ay_0>1\cap A<0\}\\ &= \left\{A|A<\frac{1}{y_0}\right\}\end{aligned}$$

which completes the proof of the first assertion. The second statement follows in largely a similar manner. In this case, $y(t) \to \infty$ if and only if $A < 0$ and $y_0 > \frac{1}{A}$. If $y_0 > 0$, the second inequality always holds provided $A < 0$. On the other hand, if $y_0 < 0$, then

$$\begin{aligned}\left\{A|A<0\cap y_0>\frac{1}{A}\right\} &= \{A|Ay_0<1\cap A<0\}\\ &= \left\{A|\frac{1}{y_0}A<0\right\}\end{aligned}$$

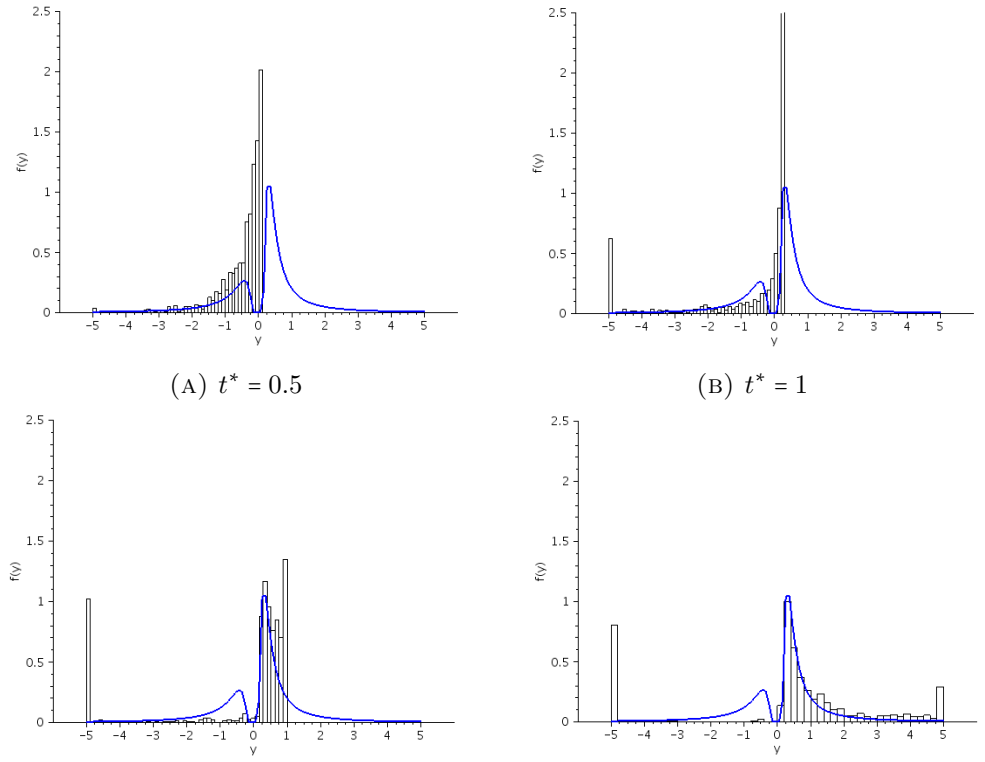
□

Using this proposition, we can now approximate the percentage of observations represented by the two bars. In our specific example, $y_0 = -1$ and $A \sim N(1, 4)$, so $\mathbb{P}(A < \frac{1}{y_0}) = \mathbb{P}(A < -1) \approx 0.1587$ and $\mathbb{P}(\frac{1}{y_0} < A < 0) = \mathbb{P}(-1 < A < 0) \approx 0.1499$. As the sample size $n = 1000$ is quite large, the histogram theorem (Proposition 1.8) applies, so we expect close to 15.9% of the observations on the left bar and nearly 15% on the right bar. After taking the average percentage for five samples of size 1000, we obtained 16.8% for the left bar and 16.7% for the right bar, which compares favorably with our estimates.

Looking back at Figure 2.2 once more, notice that the observations outside the two bars closely match the distribution of the positive part of the reciprocal Gaussian. This is not completely surprising since we have seen that $y(t)$ converges to $1/A$ as $t \to \infty$. Unfortunately, we are unable to state a precise probabilistic statement that describes this observation. We know from the histogram theorem (Proposition 1.8) that the probability of each class in the histogram converges in probability to the corresponding area under the distribution of $1/A$. However, the distribution of Y converges to the positive half of $1/A$ only when t^* is large enough. The difficulty of transforming Proposition 2.1 into a precise result about the law of $y(t^*)$ is in stating precisely how large both the sample size n and time t^* should be *simultaneously*. Indeed, the law of $y(t^*)$ when θ is random is well-estimated by a sample only when t^* is large enough but has no limit when t^* tends to $+\infty$. This problem is true in general even if we choose other values of y_0, μ, σ, or b.

REMARK 2.2. *This value of t^* for which the distribution of the small values becomes close to that of the positive half of the reciprocal Gaussian distribution is*

not that large. For example, in Figure 2.3, we see that even when t^* is as small as 4, the distribution of the small values already fits the expected distribution very well.



(A) $t^* = 0.5$ (B) $t^* = 1$

(C) $t^* = 2$ (D) $t^* = 4$

FIGURE 2.3. Histogram of y_i, where $y_0 = -1$ and $A \sim N(1, 4)$ for various values of t^*, with the graph of the pdf of the reciprocal Gaussian distribution included for comparison. In all cases, $y_{hi} = 5$, $y_{lo} = -5$, and the sample size is $n = 1000$.

2.2. An example in the Riccati case

So far, we have seen that even in the very basic case of a linear differential equation, one often needs to resort to Monte Carlo simulations to approximate the law of $y(t^*)$. We shall now show that, unfortunately, such simulations may be ineffective to find the law of $y(t^*)$ in certain differential equations.

Suppose we wish to study the law of the sample of solutions at time t^* of a Riccati equation $x' = Ax^2 + Bx + C$, where the coefficients A, B, and C are either random variables or constants. One natural way to do this would be to repeatedly sample A, B, and C from their corresponding distributions using some statistical software, and then to compute the solution at a specific time t^*, as in the previous section. While this procedure looks simple, the behavior

of the solution of Riccati equations can easily cause problems when computing these solutions numerically, as we shall now show in a specific example.

EXAMPLE 2.3. Consider the logistic differential equation $y' = -Ry(1-y)$, where the initial point is $y(0) = 2$, and R is Gaussian with mean 1 and standard deviation 2. As before, we use Scilab to generate a sample of size 1000 and calculate the solution at the times $0, 0.01, 0.02, ..., 1$. Unfortunately, we will most probably be unable to generate the 1000 trajectories, because we will get an error of the following form:

```
lsoda-- at t (=r1), mxstep (=i1) steps
needed before reaching tout
where i1 is :  500
where r1 is :  0.2739585931588D+00
Warning:  Result may be inaccurate.
```

The problem is due to the explosion of some solutions of the differential equation $y' = -Ry(1-y)$ in finite time. To understand why this occurs, consider for example, the case where $R > 0$ and $y_0 = y(0) > 1$. Straightforward integration gives us the solution of the differential equation as

$$y(t) = \frac{1}{1 + a_0 e^{Rt}} \tag{2.3}$$

where $a_0 = \frac{1-y_0}{y_0}$. Since $y_0 > 1$, a_0 is clearly negative. Then, as t approaches the positive value $\bar{t} = \frac{1}{R}\ln(-\frac{1}{a_0})$, the denominator of (2.3) approaches 0, and so $\lim_{t\to\bar{t}} y(t) \to +\infty$. (Similarly, it is not difficult to verify that if $R < 0$ and $y_0 < 0$, then $\lim_{t\to\bar{t}} y(t) \to -\infty$.) Note that this value of $\bar{t}$ could be quite small. For example, if $R \approx -2.5301136$, then $\bar{t} \approx 0.274$, as given in the error message in the previous example. In fact, any $R < -\ln 2 \approx -0.69$ will result in an explosion at a time $\bar{t} < 1$. Under the assumption that $R \sim N(1, 4)$, this has a 20% probability in every sample. Thus, even if we are only interested in the histogram of the values of the solution at a certain time t^*, we will most likely be unable to obtain this, due to the existence of these poles.

In the case of a Riccati equation, there is a way to overcome this problem: each solution may be extended to $+\infty$ or $-\infty$. To see this, one can make a change of manifold and take a look at two maps instead of one. Suppose that we wish to study the general differential equation

$$y' = Ay^2 + By + C. \tag{2.4}$$

Suppose that the differential equation has two real constant solutions α_1 and α_2. Then we can rewrite (2.4) as $y' = A(y-\alpha_1)(y-\alpha_2)$. Now let $z = \frac{1}{y-\alpha_1}$. Then

$y = \alpha_1 + \frac{1}{z}$, and

$$\begin{aligned} z' &= -\frac{1}{(y-\alpha_1)^2}y' \\ &= -z^2A(y-\alpha_1)(y-\alpha_2) \\ &= -z^2A\frac{1}{z}(\frac{1}{z}+(\alpha_1-\alpha_2)) \\ &= -A+\beta z, \end{aligned}$$

where $\beta = A(\alpha_1 - \alpha_2)$. Here, $y(t)$ and $z(t)$ are two expressions of the same trajectory in the two charts.

To implement the two charts numerically (see Example 2.6 below), we can set an upper bound U and a lower bound L, and suppose that the second chart is produced using the mapping $z = 1/(y - \alpha_1)$. Starting from the logistic equation, suppose that the solution reaches a value of U at time t_1. Since the graph would typically grow to infinity, we switch to the corresponding linear differential equation, where it must satisfy $z(t_1) = 1/U$. This transformation avoids the problem of $y(t)$ going to infinity in finite time because as $y(t) \to \infty$, $z(t)$ remains defined, and decreases instead to 0. A similar procedure can be applied when $y(t)$ goes to $-\infty$, where we move to the linear equation when the trajectory crosses the lower bound L. Thus, the method allows us to avoid the errors brought about by the pole in finite time of the Riccati equation by converting the equation to its linear version. To combine the results into one histogram, every time we switch to the second map and obtain $z(t^*)$, we simply store the corresponding value in the original map, which is $\alpha + 1/z(t^*)$.

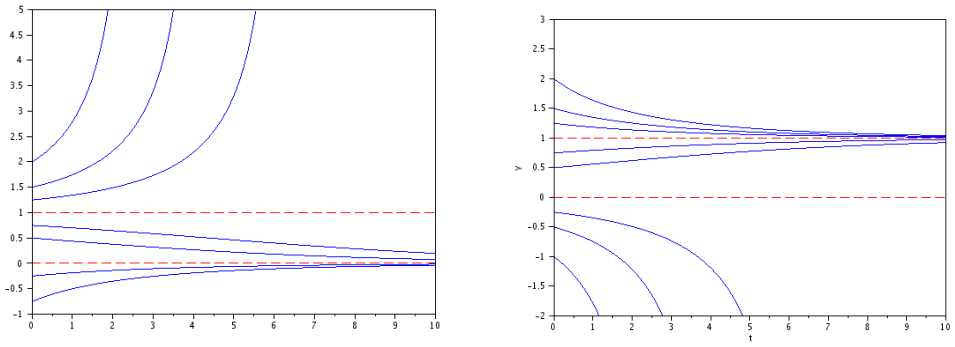
Geometrically, this corresponds to a change of chart which transforms $\mathbb{R}^2$ into a cylinder, where we join $y = +\infty$ and $y = -\infty$ together. This means that once we "reach" $+\infty$, we will be able to continue, but now passing through negative values.

EXAMPLE 2.4. Going back to the equation $y' = -Ry(1 - y)$ in Example 2.3, the corresponding linear differential equation based using the constant solution $\alpha_1 = 0$ is $z' = Rz - R$, where R has a Gaussian distribution with mean μ and variance σ^2, and $z = 1/y$. The solution of the converted differential equation is

$$z(t) = (z_0 - 1)e^{Rt} + 1. \tag{2.5}$$

As before, due to the exponential nature of the solution, the distribution of $y(t^*)$ quickly resembles the behavior when $t \to \infty$. Thus, provided t^* is large enough, we expect the resulting distribution to depend on the value of R. Looking at the direction fields of our differential equation (see Figure 2.4), we see that if $R > 0$, $y(t)$ goes to $+\infty$ as $t \to \infty$. These will all then converge to 0 through negative values after we apply our transformation. Furthermore, if $R < 0$, $y(t)$

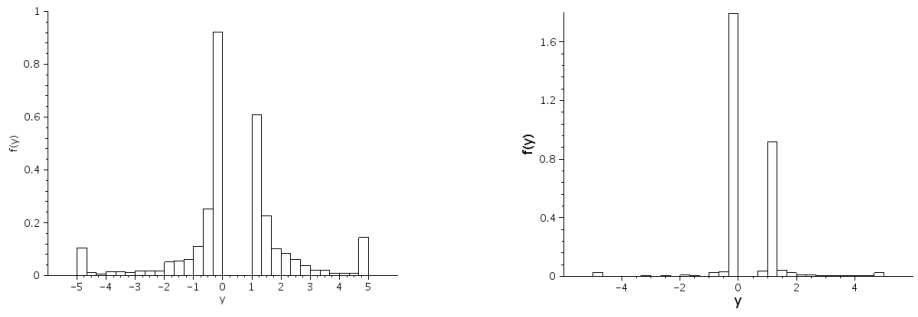
converges to 1 from above. Thus, we would expect the distribution of $y(t^*)$ to consist of values which are concentrated just below 0 and above 1, with nothing in the interval $(0, 1)$.



(A) $R > 0$ (B) $R < 0$

FIGURE 2.4. Some solutions of $y' = -Ry(1-y)$. Here, we show the two possible cases, depending on the sign of R. In the first case, $R = 0.25 > 0$, while in the second, $R = -0.25 < 0$.

Our intuition is validated when we construct the histogram of y, as shown in Figure 2.5. We constructed two histograms, one when $t^* = 1$, while the other is when $t^* = 5$. In our original case, since the chosen $t^* (= 1)$ is quite small, there remains a good percentage of values which are to the much greater than 1 or much less than 0, as seen in Figure 2.5a. These represent those values which have not yet reached the limit, which is either 0 or 1. As t^* increases, the histogram quickly approaches that of two Dirac masses at 0 and 1. In Figure 2.5b, we see that almost all the values are close to 0 and 1.

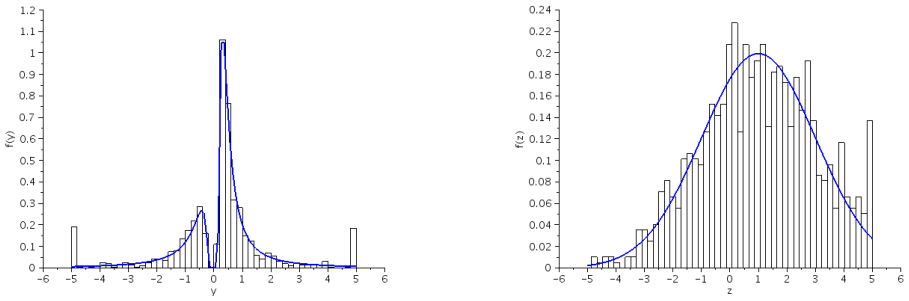


(A) Histogram when $t^* = 1$. (B) Histogram when $t^* = 5$.

FIGURE 2.5. Histogram of $y(t^*)$, where the initial point is $y_0 = 2$. All values greater than $y_{hi} = 5$ and less than $y_{lo} = -5$ are set to 5 and -5, respectively.

In the next example, we shall show that the reciprocal Gaussian distribution that we introduced in Section 1.1.3 can also appear in the context of a Riccati equation.

EXAMPLE 2.5. Consider the distribution of the value of the solution at time t^* of $y' = -Ry^2 + y$, where $R \sim N(1, 4)$ and $y(0) = y_0 < 0$. Then, the corresponding linear differential equation after performing the reciprocal transformation is $z' = R - z$. An analysis of the direction field of the Riccati equation shows that one reaches a pole in finite time when $R > 0$ and $y_0 < 0$ or if $y_0 < 1/R$ and $R < 0$. After performing the transformation which we introduced above, the resulting value at $y(t^*)$, where t^* is large, for these two cases can be shown to converge to $1/R$. If $y_0 = -2$ and $t^* = 10$, the resulting histogram is shown in Figure 2.6:



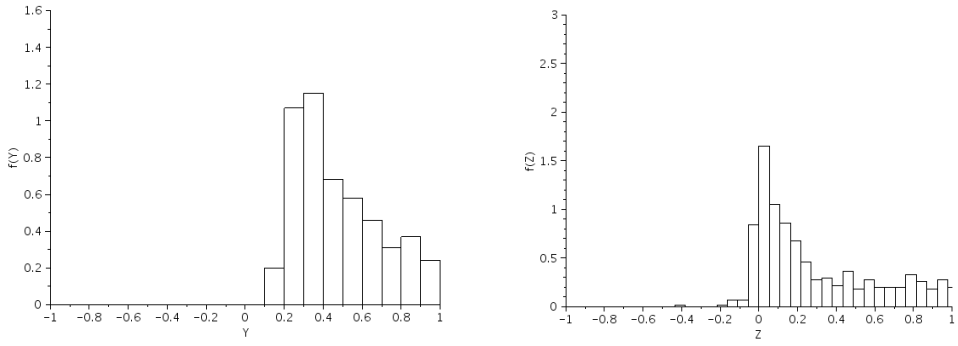
(A) Histogram of $y(t^*)$ (B) Histogram of $z(t^*)$

FIGURE 2.6. The histogram of both charts for a sample of size 1000 of $y(10)$ where $y' = -Ry^2 + y$ and $R \sim N(1, 4)$. The theoretical distribution for the reciprocal Gaussian where the normal random variable is $N(1, 4)$ is sketched for comparison.

In the above histogram, all values greater than $y_{hi} = 5$ and less than $y_{lo} = -5$ are set to 5 and -5, respectively. One can see this almost replicates the reciprocal Gaussian pdf except for the region where $1/y_0 < R < 0$, where the values of $y(10)$ are known to converge to 0 as $t \to \infty$. Here $T = 10$, which is already quite large. However, if y_0 is large negative (say, $y_0 = -20$), the probability of this region is small, and so we obtain a pdf which is very close to that of the reciprocal Gaussian distribution.

Clearly, there is an overlap in the information provided by the two graphs in Figure 2.6. However, we can avoid this by choosing appropriate intervals in the two charts. For example, if we take the histogram of $y(t^*)$ over the interval (a_1, a_2), where $0 < a_1 < a_2$, the values for the intervals $(-\infty, a_1) \cup (a_2, +\infty)$ is captured by taking the histogram of $z(t^*)$ over $(1/a_2, 1/a_1)$. Thus, taking two charts can be thought of as another way to "see" the values which are spread out (big positive or negative values).

EXAMPLE 2.6. In the previous section, we examined the distribution of $y(t^*)$ for the linear differential equation $y' = -Ay + 1$. The corresponding Riccati equation is $z' = -z^2 + Az$. The approximate distribution was constructed by "cutting" the histogram by setting all values beyond $y_{hi} = 5$ and $y_{lo} = -5$ to 5 and -5, respectively. The idea of two charts which we have just discussed can be used as an alternative to show the full results. In particular, Figure 2.7 displays y over $(-1, 1)$ and the corresponding *quadratic* z also over $(-1, 1)$, which, when combined, show a "complete picture" of the distribution of Y on a circle.



(A) Histogram of $y(t^*)$ for $(-1, 1)$ (B) Histogram of $z(t^*)$ for $(-1, 1)$

FIGURE 2.7. The histogram of both charts for a sample of size 1000 of $y(5)$ where $y' = -Ay + 1$ and $A \sim N(1, 4)$.

2.3. Polynomial Chaos

In the previous section, we have already seen several ways to study the distribution of the solution at time t^* of a system of differential equations with random coefficients. These have mainly relied on direct Monte Carlo simulations and some workarounds to be able to display the histogram of the results properly. However, the idea to use a two chart representation to study the law with a Monte Carlo approach even when some solutions explode in finite time (and thus before t^*) which is helpful for Riccati equations will no longer be possible for more general equations or systems. Thus there is a need to look at other approaches when Monte Carlo simulation no longer works.

In this section, we explore another approach, which involves constructing *expansions* to approximate the law of the solutions at time t^*. This will use the concept of polynomial chaos, which shall provide us with basis random variables to approximate a given random variable. The first section gives the basic concepts of generalized polynomial chaos (gPC) expansions, as described by Xiu and Karniadakis in their paper [**48**], as well as in the subsequent book by Xiu [**49**]. The second part explains how these expansions can be used to approximate the law that we are looking for in this chapter.

2.3.1. Introduction and Definitions.

DEFINITION 2.7. *Let $\mathcal{V}$ be the Euclidean vector space of square integrable random variables normed by $||V||^2 = \mathbb{E}(V^2)$ for all $V \in \mathcal{V}$. Let $Z \in \mathcal{V}$ be a given random variable with finite moments. A generalized Z-polynomial chaos (gPC) is a sequence of random variables $\Phi_0(Z), \Phi_1(Z), ..., \Phi_k(Z), ...$, where $\Phi_0, \Phi_1, ..., \Phi_k, ...$ are orthogonal polynomials of degree 0,1,...,k,... respectively. That is, we have*

$$\mathbb{E}(\Phi_j(Z)\Phi_k(Z)) = \gamma_k \delta_{jk}$$

where δ_{jk} is the Kronecker delta, which is 1 if $j = k$ and 0 otherwise, and $\gamma_k = \mathbb{E}(\Phi_k^2(Z))$ are positive constants.

It is customary to normalize the elements in Γ_p in some manner, so as to obtain a unique gPC of each order $j \leq p$. There are many ways to do this. In principle, we can construct an orthonormal basis by requiring that $\mathbb{E}(\Phi_j^2(Z)) = 1$ for each j. A second option would be to set the leading coefficient of Φ_j to be 1. In both cases, this would define a unique gPC basis function of order p. We shall choose the second option, and so from this point on, whenever we mention the gPC basis function $\Phi_p(Z)$, it shall refer to the element of Γ_p which has a leading coefficient of 1.

EXAMPLE 2.8. Depending on what distribution we choose for the given random variable Z, we obtain a different set of gPC basis polynomials. Table 2.1 lists the gPC basis random variables of degrees 0 to 3 where Z is either $N(0,1)$ or $U(-1,1)$.

TABLE 2.1. The generalized polynomial chaos basis random variables of degrees 0 to 3

Degree	Distribution of Z:	
	$N(0,1)$	$U(-1,1)$
$\Phi_0(Z)$	1	1
$\Phi_1(Z)$	Z	Z
$\Phi_2(Z)$	$Z^2 - 1$	$Z^2 - \frac{1}{3}$
$\Phi_3(Z)$	$Z^3 - 3Z$	$Z^3 - \frac{3}{5}Z$

To obtain the basis given above, we can proceed in a recursive manner by using a series of orthogonalization procedures. Assume first that $Z \sim N(0,1)$. Clearly, $\Phi_0(Z) = 1$. Then $\Phi_1(Z)$ is a linear function $Z + a$ such that $\mathbb{E}(Z + a) = 0$ (by orthogonality with $\Phi_0(Z)$). This implies that $a = 0$, so $\Phi_1(Z) = Z$. Next, $\Phi_2(Z)$ is a quadratic function $\Phi_2(Z) = Z^2 + bZ + a$ such that it is orthogonal to *both* $\Phi_0(Z)$ and $\Phi_1(Z)$. A straightforward computation leads to $a = -1$ and $b = 0$, so $\Phi_2(Z) = Z^2 - 1$. All the higher order basis functions can be computed

in a similar manner. A similar procedure but using $Z \sim U(-1,1)$ gives us the random variables in the third column. □

REMARK 2.9. *The gPC basis functions above are, in fact, of the form of classical orthogonal polynomials. For example, for $Z \sim N(0,1)$, the function Φ_k are the Hermite polynomials, while for $Z \sim \mathcal{U}(-1,1)$, they are the Legendre polynomials.*

REMARK 2.10. *It also follows from the definition of the gPC basis functions that $\mathbb{E}(\Phi_k(Z)) = 0$ for $k > 0$. This follows because $\mathbb{E}(\Phi_k(Z)) = \mathbb{E}(\Phi_0(Z) \cdot \Phi_k(Z)) = 0$. Thus the $\Phi_k(Z)$ are all centered random variables for $k > 0$.*

2.3.2. Approximation using polynomial chaos. For the succeeding discussion, we shall focus only on the polynomial chaos basis functions produced when Z has a standard normal distribution. The main interest in gPC is that they can be used to approximate random variables Y.

Let Y be an L^2-integrable with known distribution. We define the Nth order gPC orthogonal projection, or the *Nth order gPC expansion of Y* as

$$P_N Y = \sum_{k=0}^{N} y_k \Phi_k(Z) \tag{2.6}$$

where

$$y_k = \frac{1}{\gamma_k}\mathbb{E}[Y\Phi_k(Z)] = \frac{\mathbb{E}[Y\Phi_k(Z)]}{\mathbb{E}[\Phi_k^2(Z)]} \tag{2.7}$$

are known as the *modes* of the expansion.

The convergence properties of these expansions are very similar to that of the classical Fourier approximation. For functions f belonging to $L^2([-\pi,\pi])$, we know that the approximation using the Fourier basis converges to f in mean-square. While these gPC expansions approximate random variables and not functions, it turns out that they retain a similar convergence property. In fact, it can be shown that the orthogonal projection defined above converges in mean-square to Y in the case where $Y = f(Z)$ is a function of the random variable Z on which the polynomial chaos is built. That is, in this case, we have $\|Y - P_N Y\| \to 0$ as $N \to \infty$, where the norm is the standard mean-square norm $\|Y\|^2 = \mathbb{E}(Y^2)$. This convergence follows directly from the corresponding result for real functions. For a proof, we refer to Theorem 6.2.3 in [**10**] in the case where f is bounded and [**5**] for the unbounded case.

To see an example on how to construct such an expansion, we will now approximate a lognormal random variable using a Gaussian gPC.

PROPOSITION 2.11. *Let $Y = e^Z$ be a lognormal random variable, where $Z \sim N(0,1)$. Then the polynomial chaos expansion for Y is given by*

$$Y = e^{1/2} \sum_{k=0}^{\infty} \frac{1}{k!} H_k(Z),$$

where $H_k(Z)$ is the Hermite polynomial of order k in the variable Z.

Proof. We have already explained in Example 2.8 that when Z is gaussian, the Z-gPC is the family $H_k(Z)$ of Hermite polynomials of order k. In this proof, we will only show that the coefficient for the first four terms is true by constructing the polynomial chaos expansion of order 3. The remaining terms can be obtained in a similar manner.

To do this, as the polynomial chaos bases of orders 0 to 3 are the ones given in the second column of Table 2.1, we want to show that

$$P_3 Y = e^{1/2}\left[1 + Z + \frac{1}{2}(Z^2 - 1) + \frac{1}{6}(Z^3 - 3Z)\right].$$

For this, note that $Y = \sum_{k=0}^{4} y_k \Phi_k(Z)$. To find the coefficients of the expansion, we need to apply (2.7).

First, we have

$$y_0 = \mathbb{E}(Y) = e^{1/2}$$

as the expectation of e^Z, where $Z \sim N(0,1)$ is $e^{1/2}$.

Furthermore,

$$y_1 = \frac{\mathbb{E}(YZ)}{\mathbb{E}(Z^2)} = \frac{\mathbb{E}(e^Z Z)}{\mathbb{E}(Z^2)} = e^{1/2}$$

since

$$\begin{aligned}
\mathbb{E}(e^Z Z) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} z e^z e^{-\frac{1}{2}z^2} dz \\
&= \frac{e^{1/2}}{\sqrt{2\pi}} \int_{-\infty}^{\infty} z e^{-\frac{1}{2}(z-1)^2} dz \\
&= e^{1/2} \underbrace{\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} z e^{-\frac{1}{2}(z-1)^2} dz}_{\text{mean of } N(1,1)} \\
&= e^{1/2}.
\end{aligned}$$

To compute y_2:

$$y_2 = \frac{\mathbb{E}(Y\Phi_2(Z))}{\mathbb{E}((Z^2-1)^2)} = \frac{\mathbb{E}(e^Z(Z^2-1))}{\mathbb{E}(Z^4 - 2Z^2 + 1)}$$

Here, the numerator can be computed as follows:

$$\mathbb{E}(e^Z(Z^2-1)) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (z^2-1) e^{1/2} \cdot e^{-\frac{1}{2}(z-1)^2} dz$$

$$= e^{1/2}\left[\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}}z^2e^{-\frac{1}{2}(z-1)^2}dz - \int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(z-1)^2}dz\right]$$
$$= e^{1/2}$$

In the second to the last line, the first term is equal to 2 since it is the same as $\mathbb{E}(\widetilde{Z}^2) = Var\widetilde{Z} + \mu^2$ where $\widetilde{Z} \sim N(1,1)$, while the second term is 1 since it is the mean of a $N(1,1)$ random variable. On the other hand, the denominator is 2 since $\mathbb{E}((Z^2-1)^2)) = \mathbb{E}(Z^4 - 2Z^2 + 1) = 3 - 2 + 1 = 2$. Thus, $y_2 = \frac{1}{2}e^{1/2} = \frac{1}{2!}e^{1/2}$.

Finally, to compute y_3, we have

$$y_3 = \frac{\mathbb{E}(Y\Phi_3(Z))}{\mathbb{E}(\Phi_3^2(Z))}$$

The denominator is equal to 6 since

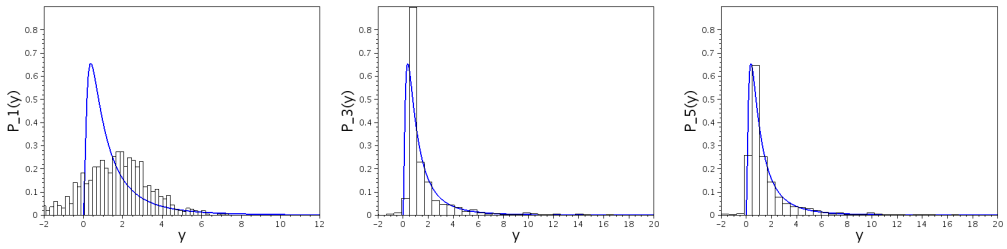
$$\mathbb{E}(\Phi_3^2) = \mathbb{E}((Z^3-3Z)^2) = \mathbb{E}(Z^6 - 6Z^4 + 9Z^2) = 15 - 18 + 9 = 6$$

On the other hand, the numerator can be computed as follows:

$$\begin{aligned}\mathbb{E}(e^Z(Z^3-3Z)) &= e^{1/2}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}}(z^3-3z)e^{-\frac{1}{2}(z-1)^2}dz \\ &= e^{1/2}[\mathbb{E}(\widetilde{Z}^3) - 3\mathbb{E}(\widetilde{Z})], \text{where } \widetilde{Z} \sim N(1,1) \\ &= e^{1/2}.\end{aligned}$$

In the last step, we used the fact that the third non-central moment of a Gaussian random variable with mean μ and variance σ^2 is $\mu^2 + 3\mu\sigma^2$, and so for $\mu = \sigma = 1$, is equal to 4. Thus, $y_3 = \frac{1}{6}e^{1/2} = \frac{1}{3!}e^{1/2}$. □

Figure 2.8 gives a comparison between the pdf of a lognormal random variable $f(Z) = e^Z$ and the histogram of the pth order polynomial chaos expansion of $Y = f(Z)$. We can see that while the histogram of the first-order expansion does not capture the shape of the pdf very well, the higher-order expansions quickly become more accurate approximations of the true pdf.



(A) Order 1 (B) Order 3 (C) Order 5

FIGURE 2.8. Comparison of the lognormal pdf with a histogram of its polynomial chaos expansion of successive orders 1, 3, and 5.

As we have seen in the example above, if Y can be written as a function of Z, the computation of its coefficients is rather straightforward. However, in most cases, the only thing we know is the distribution of the random variable in question. In this case, one cannot compute $\mathbb{E}(Y \cdot \Phi_k(Z))$ in the numerator of $\hat{f}_k$ because the dependence between Y and Z is unknown. It is still possible to construct a gPC expansion of Y, but with slightly weaker convergence properties. Let F_X represent the cdf of the random variable X, and denote by I_X the support of X. Then, recall from Proposition 1.11 that Y has the same distribution as $F_Y^{-1}(F_Z(Z))$. We can then rewrite the expression for y_j in (2.7) as follows:

$$y_j = \frac{\mathbb{E}_Z[F_Y^{-1}(F_Z(Z))\Phi_j(Z)]}{\mathbb{E}[(\Phi_j(Z))^2]}$$

Note that while Y and $F_Y^{-1}(F_Z(Z))$ have the same distribution, they are not the same random variables. Thus, unlike the mean-square convergence in the previous case, the best result we can obtain on $P_N f$ in this case is that it converges in probability to f.

PROPOSITION 2.12 (Theorem 5.7 in [**49**]). *Let Y be a random variable with cdf $F_Y(y)$ and assume $\mathbb{E}(Y^2)$ is finite, and let Z be a random variable with cdf $F_Z(z)$, and finite moments such that its gPC basis functions exist with $\mathbb{E}[\Phi_m(Z)\Phi_n(Z)] = \delta_{mn}\gamma_n$ for all $m, n \in \mathcal{N}$. Let*

$$Y_N = \sum_{k=0}^{N} a_k \Phi_k(Z) \tag{2.8}$$

where

$$a_k = \frac{1}{\gamma_k}\mathbb{E}_Z[F_Y^{-1}(F_Z(Z))\Phi_k(Z)], \quad 0 \le k \le N \tag{2.9}$$

Then Y_N converges to Y in probability.

Proof. Denote by $\tilde{Y}$ the function $G(Z) = F_Y^{-1}(F_Z(Z))$. By Proposition 1.11, $\tilde{Y}$ has the same probability distribution as that of Y, and so must have a finite second moment as well. Thus,

$$\begin{aligned}
\mathbb{E}[\tilde{Y}^2] &= \int_{I_Y} y^2 f(y) dy \\
&= \int_0^1 (F_Y^{-1}(u))^2 du \\
&= \int_{I_Z} (F_Y^{-1}(F_Z(z)))^2 f(z) dz,
\end{aligned}$$

which is finite. Thus, $\tilde{Y}$ is a mean-square integrable function of Z. Since (2.8) is in fact the orthogonal projection of $\tilde{Y}$ using the Nth-degree gPC basis, Y_N converges in mean square to $\tilde{Y}$. But this implies that Y_N also converges in

probability to $\tilde{Y}$, since convergence in probability follows from L^2 convergence. Since $\tilde{Y}$ and Y have the same distribution, the result follows. □

REMARK 2.13. *Polynomial chaos expansions can also be used to approximate the means and variances of random variables. In particular, we have for any* $N \geq 0$,

$$\mathbb{E}(P_N Y) = y_0 \quad \text{and} \quad Var(P_N Y) = \mathbb{E}\left[\sum_{j=1}^{N} y_j^2 \Phi_j^2(Z)\right] \tag{2.10}$$

To see this, note that

$$\mathbb{E}(P_N Y) = \mathbb{E}\left[\sum_{i=0}^{N} y_i \Phi_i(Z)\right] = y_0,$$

since the expectation of $\Phi_i(Z)$ *for* $i \geq 1$ *is 0 from Remark 2.10. On the other hand, to compute the approximate variance, we have*

$$\begin{aligned}
Var(P_N Y) &= \mathbb{E}[(P_N Y - \mathbb{E}(P_N Y))^2] \\
&= \mathbb{E}\left[\left(\sum_{j=0}^{N} y_j \Phi_j(Z) - y_0\right)^2\right] \\
&= \mathbb{E}\left[\left(\sum_{j=0}^{N} y_j \Phi_j(Z)\right)^2 - 2y_0 \sum_{j=0}^{N} y_j \Phi_j(Z) + y_0^2\right] \\
&= \mathbb{E}\left[\sum_{j=0}^{N} y_j^2 \Phi_j^2(Z) + 2\sum_{i \neq j} y_i y_j \Phi_i(Z)\Phi_j(Z) - 2y_0 \sum_{j=0}^{N} y_j \Phi_j(Z) + y_0^2\right] \\
&= y_0^2 + \mathbb{E}\left[\sum_{j=1}^{N} y_j^2 \Phi_j^2(Z)\right] - 2y_0^2 + y_0^2 \\
&= \mathbb{E}\left[\sum_{j=1}^{N} y_j^2 \Phi_j^2(Z)\right]
\end{aligned}$$

2.3.3. Application to a system of differential equations with random coefficients. We now describe how we can use polynomial chaos expansions to estimate the law of the solution at t^*, $y(t^*)$, of a differential equation with random coefficients. First, we shall explain the principle of the method and then illustrate how it works on two examples.

Consider the differential system $y' = g(y;\theta)$ where θ consists of one or more random coefficients. Let $y(t;\theta)$ be the solution and assume that θ is a function $f(Z)$ of a given random variable Z having finite moments. Since for any t, $y(t;\theta)$ will, in effect, be a random variable, then we can construct the Z-polynomial chaos expansion of $y(t;\theta)$

$$P_N y(t;\theta) = \sum_{i=0}^{\infty} y_i(t,\theta)\Phi_i(Z), \tag{2.11}$$

where $\Phi_i(Z)$ is the ith-degree Z-gPC. Indeed, for all t for which the solution $y(t;\theta)$ exists, this random variable is a function of θ and thus a function of Z.

Also, we can construct the Z-gPC expansion of θ:

$$\theta = \sum_{i=0}^{\infty} \theta_i \Phi_i(Z) \tag{2.12}$$

The idea of the method is the following: if it is possible to compute the modes $y_i(t^*)$ of $y(t^*;\theta)$, then the pdf of the sequence

$$y_0(t^*),\ y_0(t^*) + y_1(t^*)\Phi_1(Z),\ y_0(t^*) + y_1(t^*)\Phi_1(Z) + y_2(t^*)\Phi_2(Z),\ \ldots$$

will give a sequence of pdfs that approximates better and better the pdf of the random variable $y(t^*;\theta)$ we are interested in.

The computation of the coefficients $y_i(t)$ is straightforward, even if it usually involves heavy computation. From the differential equation, we have

$$\frac{d}{dt}\left(\sum_{i=0}^{\infty} y_i(t)\Phi_i(Z)\right) = g\left(\sum_{i=0}^{\infty} y_i(t)\Phi_i(Z); \sum_{i=0}^{\infty} \theta_i \Phi_i(Z)\right).$$

After term by term differentiation of the left-hand side and Taylor expansion of g on the right-hand side, we end up, after performing a projection of the above equation onto each element of the basis $\{\Phi_i(Z)\}$, with a system of differential equation having the different modes $y_i(t)$ as variables.

EXAMPLE 2.14. Consider the linear differential equation $y' = -Ay$, where A is assumed to be lognormal based from a $N(0,1)$ random variable, and the initial point $y(0) = 1$. In this case, $y(t^*) = (y(0))e^{-At^*}$, whose law we can compute explicitly. Also, since all the values of A are positive, we shall not encounter the problems encountered in the section 2.1. We shall only take the second order polynomial chaos expansions of A and y using the Hermite polynomials. That is, we approximate A by $\sum_{i=0}^{2} a_i \Phi_i(Z)$ and $y(t)$ by $\sum_{i=0}^{2} y_i(t)\Phi_i(Z)$.

From Proposition 2.11, we know the value of the coefficients a_i in the expansion for A. In particular, the second-order gPC expansion of A is given by

$$P_2 A = e^{1/2}(\Phi_0(Z) + \Phi_1(Z) + \frac{1}{2}\Phi_2(Z)).$$

In addition, the initial condition has a trivial gPC expansion, where $y_0(0) = 1$ and $y_i(0) = 0$ for all $i \geq 1$.

If we replace A and y by their corresponding second order gPC expansions, we obtain

$$\frac{d}{dt}\left[\sum_{i=0}^{2} y_i(t)\Phi_i(Z)\right] = -\left[e^{1/2}\left(\Phi_0(Z) + \Phi_1(Z) + \frac{1}{2}\Phi_2(Z)\right)\right]\left[\sum_{i=0}^{2} y_i(t)\Phi_i(Z)\right].$$

Expanding and taking the projection with each basis $\Phi_j(Z)$, $j = 0, 1, 2$, we obtain the following differential system satisfied by $y_0(t), y_1(t), y_2(t)$:

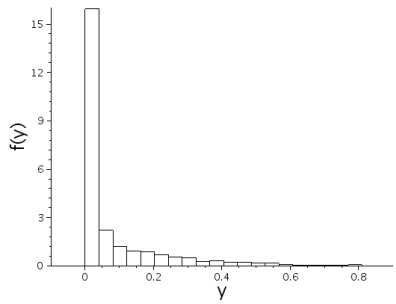
$$\text{(2.13)} \qquad \begin{cases} y_0'(t) &= -e^{1/2}[y_0(t) + y_1(t) + y_2(t)] \\ y_1'(t) &= -e^{1/2}[y_0(t) + 2y_1(t) + 2y_2(t)] \\ y_2'(t) &= -e^{1/2}[y_0(t) + 2y_1(t) + 6y_2(t)] \end{cases}$$

Notice that the resulting system is, in fact, a system of coupled linear differential equations, which can either be solved analytically or using any computational program.

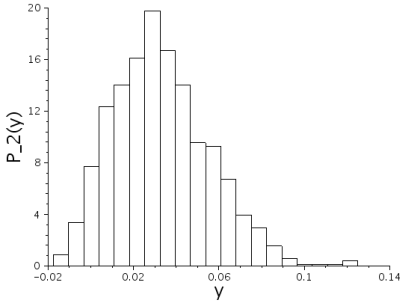
Figure 2.9 compares the resulting histogram of 1000 simulations of the value of the solution of the differential equation at $t^* = 5$ with 1000 simulations of the gPC approximation of the solution evaluated at $t^* = 5$. As before, notice that while the accuracy of the approximation is not that good when the order of the expansion is low (for example, when it is degree 2), it quickly converges to the true histogram as the order increases.

The main weakness of the gPC method in computing an expansion of the law of $y(t^*)$ is the computational issue. Even with just a single random coefficient and just using second-order polynomial chaos expansions, one can already end up with a rather complicated system like (2.13) to be solved. Indeed, it is easy to see that this system in the modes $y_i's$ quickly increases in dimension as the number of equations and random coefficients increases. Also, when there are multiple random coefficients, one will need to incorporate in their gPC expansions the correlations between these coefficients. Thus, one will often need to use a very low order gPC expansion for each random coefficient, or include certain simplifying assumptions even to assume that the unknown random coefficients are independent. The good news is that, when the computation of the first modes $y_i(t)$ is tractable, even a low order expansion already produces quite a good approximation.

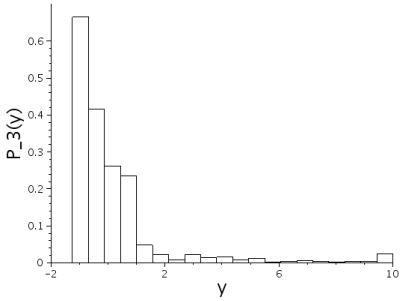
EXAMPLE 2.15. To have an idea on how the computation may become rather heavy, we now consider a second example. Stanescu and Charpentier [**42**] show how to use polynomial chaos expansions to approximate the solution of a simplified Monod model of microbial growth. In a Monod model, the rate of growth depends on the amount of necessary nutrients. There are at least two differential equations, one for the microorganism population, and another for the amount of each nutrient. If there is an abundant supply of nutrients, the model resembles that of an exponential growth model. As the amount of available nutrients decreases, then so does the growth rate as well. This is usually used to model



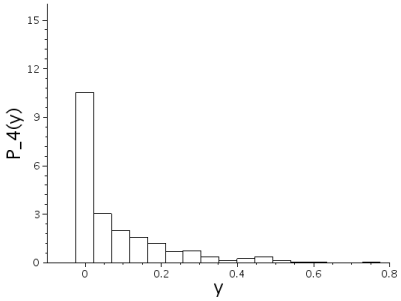
(A) From the original differential system



(B) Order 2



(C) Order 3



(D) Order 4

FIGURE 2.9. Comparison of the pdf of $y(5)$ with a histogram of the polynomial chaos expansion of various orders. The upper left graph is that of the histogram of $y(5)$, simulated directly. Graphs (B), (C), (D) represent the corresponding approximate histograms of the gPC expansions of $y(5)$ of orders 2, 3, and 4, respectively. For each histogram, the horizontal axis represents the value of $y(5)$ while the vertical axis gives the corresponding relative frequency.

the growth of microorganisms in test tubes, where there is no convection or diffusion of either microbes or nutrients. Under certain simplifying assumptions[1], the resulting system is given as follows:

$$\begin{cases} \dfrac{dx}{dt} &= \dfrac{\mu_x y}{K+y} x \\ \dfrac{dy}{dt} &= \dfrac{\mu_y y}{K+y} x \end{cases} \tag{2.14}$$

[1]The assumptions include the following: first, that there is just a single necessary nutrient; second, that the microbial death rate is proportional to the size of the population; and third, that the rate of microbial growth is given by the Monod kinetics reactions, $\mu(y) = \frac{\mu_x y}{K+y}$, where μ_x is the maximum specific growth rate and K is the value of y where the specific growth rate $\mu(y)$ has half its maximum value.

The variables x and y represent the mass concentration of the microbes and the soluble nutrients per unit volume, respectively. The coefficients which are assumed to be random variables include the maximum specific growth rate μ_x, the half-growth concentration rate K, and the quantity μ_y. It will be assumed that these three coefficients are independent and functions of three iid $N(0,1)$ random variables Z_1, Z_2, Z_3.

We express each quantity in terms of its gPC expansion of order N:

$$\mu_x = \mu_{x0} + \mu_{x1}\Phi_1(Z_1) + ... + \mu_{xN}\Phi_N(Z_1)$$
$$\mu_y = \mu_{y0} + \mu_{y1}\Phi_1(Z_2) + ... + \mu_{yN}\Phi_N(Z_2)$$
$$K = K_0 + K_1\Phi_1(Z_3) + ... + K_n\Phi_N(Z_3)$$

For x and y, the gPC expansions need to consider the contributions of all three random variables, and will thus be three-dimensional chaos expansions. In this case, there will be more than one gPC basis of each order. In particular, the order n polynomial chaos basis random variables consists of all possible products of the single-variable gPC of the three random variables defined by

$$\Phi_i(Z) = \Phi_{i_1}(Z_1)\Phi_{i_2}(Z_2)\Phi_{i_3}(Z_3)$$

where $i_1 + i_2 + i_3 = n$. For example, the second-order expansion of x will have 10 terms, and have the form

$$\begin{aligned} x(t) \quad = \quad & x_0(t) + x_1(t)\Phi_1(Z_1) + x_2(t)\Phi_2(Z_2) + x_3(t)\Phi_3(Z_3) + x_4(t)\Phi_1(Z_1)\Phi_2(Z_2) \\ & + x_5(t)\Phi_1(Z_1)\Phi_3(Z_3) + x_6(t)\Phi_1(Z_1)\Phi_3(Z_3) + x_7(t)(\Phi_1(Z_1))^2 \\ & + x_8(t)(\Phi_2(Z_2))^2 + x_9(t)(\Phi_3(Z_3))^2 \end{aligned}$$

If we include all the cross-product terms between the Φ_i's, it can be shown that there will be a total of $P+1$ terms in the expansion for x and y, where

$$(P+1) = \frac{(N+3)!}{N!3!}$$

where N is the chosen order of the gPC expansion.

Denote by $\Gamma_0, \Gamma_1, ..., \Gamma_P$ a particular ordering of these $P+1$ terms. Note that the elements Γ_i, $i = 0, 1, 2, ..., P$ will depend on a different combination of random variables among Z_1, Z_2, Z_3 based on the ordering, and will not be indicated explicitly. Rewriting the first equation as

$$(K+y)\frac{dx}{dt} = \mu_x xy$$

and then substituting the PC expansions, we get

$$\left(\sum_{k=0}^{P} K_k\Gamma_k + \sum_{j=0}^{P} y_j\Gamma_j\right)\sum_{i=0}^{P}\frac{dx_i}{dt}\Gamma_i = \sum_{i=0}^{P}\sum_{j=0}^{P}\sum_{l=0}^{P} x_i y_j \mu_{xl}\Gamma_i\Gamma_j\Gamma_l.$$

As before, we shall require that the residual be orthogonal on the subspace spanned by the basis functions. To do this, we need to take the inner product of the above equation with each of the basis functions. For example, doing this with the basis function Γ_L gives

$$\sum_{i=0}^{P}\sum_{j=0}^{P}(K_k + y_j)\langle\Gamma_i\Gamma_j, \Gamma_L\rangle\frac{dx_i}{dt} = \sum_{i=0}^{P}\sum_{j=0}^{P}\sum_{l=0}^{P} x_i y_j \mu_{xl}\langle\Gamma_i\Gamma_j\Gamma_l, \Gamma_L\rangle,$$

which is a matrix equation for the vector of unknown variables $x_0, x_1, ..., x_P, y_0, y_1, ..., y_P$. One can then proceed to solve for the coefficients using a sequential method such as an explicit Runge-Kutta method.

CHAPTER 3

Estimating coefficients of systems of differential equations: a first approach

In the previous chapter, our study has focused on one aspect of the *forward problem* for differential systems, which is to describe and analyze the behavior of the state variables over time. However, the *inverse problem*, which is the estimation of the coefficients of a differential system based on known data on one or more trajectories, is not as well-studied, especially in a statistical perspective.

In this chapter, we begin to study a possible solution to the inverse problem. After briefly looking at a crude method for a logistic differential equation in Section 3.1 and two other types of coefficient estimation techniques in Section 3.2, we examine in Section 3.3 a simple, yet effective stochastic method known as *rejection sampling* to generate a sample of the coefficients. Then in Section 3.4, we give some of the properties of this method, and illustrate them using simulations. The remaining two sections, Sections 3.5 and 3.6 show how to improve the basic method, and also, how the method performs on perturbed data.

3.1. An example using a logistic model

We begin by considering a coefficient estimation problem in a single differential equation. Consider the problem of estimating the coefficients r and K in the logistic differential equation

$$y' = ry\left(1 - \frac{y}{K}\right)$$

that best fit some given data. It is possible to construct an estimate of r and K without the help of a computer. To do this, first note that we can write the logistic differential equation as

$$\frac{y'}{y} = r\left(1 - \frac{y}{K}\right). \tag{3.1}$$

We can think of this as a linear regression problem with independent variable y, slope $-r/K$, intercept r, and dependent variable y'/y. If the given data are the points $(t_1, y_1), (t_2, y_2), ..., (t_k, y_k)$, we can approximate the value of y'/y in (3.1) using a difference approximation for the derivative, which gives

$$\tau_i = \frac{y_{i+1} - y_i}{t_{i+1} - t_i} \cdot \frac{1}{y_i}, \quad i = 1, 2, ..., k-1.$$

One can then compute the regression coefficients and, consequently, r and K using the classical linear regression formulas.

EXAMPLE 3.1. Suppose we wish to fit a logistic model to the following set of data:

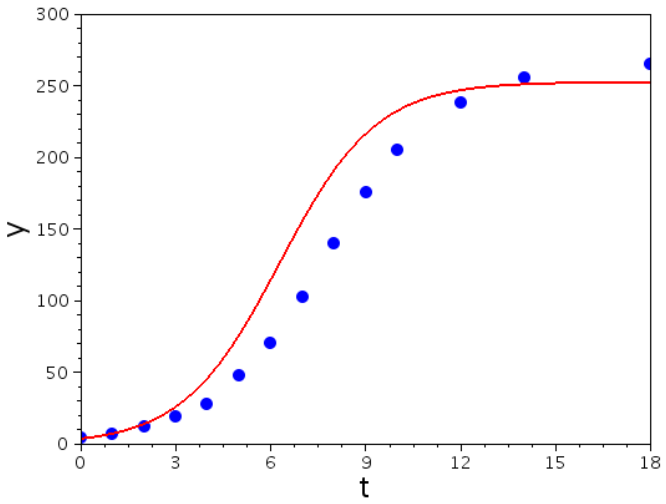
t	0	1	2	3	4	5	6	7	8	9	10	12	14	18
y	4	7	12	19	28	48	70	103	140	176	205	238	256	265

The corresponding data for the transformed model is

t	0	1	2	3	4	5	6	7	8	9	10	12	14
τ	0.75	0.71	0.58	0.47	0.71	0.46	0.47	0.36	0.26	0.16	0.08	0.04	0.01

Then the regression coefficients are $r = 0.6588$ and $-r/K = -0.0026$, so $K \approx 253$. The graph of the resulting logistic model is given in Figure 3.1.

FIGURE 3.1. Graph of the given data (represented by the o's) and the corresponding logistic curve of best fit obtained using a linear regression approach.



Based on Figure 3.1, one can see that the resulting fit is not very good. One of the main reasons for this is probably the large discrepancy between the slope of the tangent line at each time point and the difference approximation we used for the slope. Moreover, this method is difficult or even impossible to apply to more general differential systems. Fortunately, there are more advanced methods that are available to solve this problem, some of which we will review in the next section. In any of these methods, the main challenge is the estimation of the derivative (and often, even the second derivative) of the solution.

3.2. An overview of ODE coefficient estimation methods

In this section, we will give an overview of two of the most used methods to estimate the coefficients of a system of differential equations, following [**20**] and [**30**]. The first is a Newton (or quasi-Newton) method for minimizing a distance to the data. The second, a collocation method, starts by approximating the data by splines in order to compute an "approximate" derivative of the solution.

We will consider the system of differential equations of the form

$$y' = g(y;\theta) \tag{3.2}$$

with several unknown coefficients $\theta = (\theta^1, \theta^2, ..., \theta^m) \in \mathbb{R}^m$ and assume that we have known values of y at times $T = \{t_0, t_1, ..., t_k\}$, denoted $\bar{y}(T)$, where $\bar{y}(T)$ is defined as follows:

t	t_0	t_1	$\cdots$	t_k
$\bar{y}(t)$	$\bar{y}_0$	$\bar{y}_1$	$\cdots$	$\bar{y}_k$

We assume that $y \in \mathbb{R}^l$, so our differential system consists of l variables and m unknown coefficients. We shall denote by $y^i(t;\theta)$ (or simply $y^i(t)$) and y^i, $i = 1, 2, ..., l$ the ith component of $y(t;\theta)$ and $\bar{y}(T)$, respectively. Our problem is to compute the best possible coefficients θ for which the corresponding solution $y(t;\theta)$ of (3.2) fits the data $\bar{y}(T) = (t_i, y_i)_{i=0,1,...,k}$.

3.2.1. Quasi-Newton methods. Our presentation of this method is mainly based on [**28**]. Here, one uses the sum of squares of the Euclidean distance as the measure of the distance between the trajectory corresponding to a value θ^* of the coefficients in (3.2) and the known data $\bar{y}(T)$:

$$\rho(\theta^*) = \sum_{i=1}^{k} \|y(t_i;\theta^*) - y_i\|^2.$$

The objective is to determine θ for which $\rho(\theta)$ achieves its minimum value. This is done by estimating iteratively the value of θ for which the gradient of ρ is equal to the zero vector. Two ways to do this (which also work for more general problems) include Newton's method and Quasi-Newton methods.

In Newton's method, we start by assuming a starting point θ_0 in the domain. Given θ_k, we compute the next iterate as follows:

$$\theta_{k+1} = \theta_k - \alpha_k \nabla^2[\rho(\theta_k)]^{-1} \nabla\rho(\theta_k),$$

where $\nabla\rho$ and $\nabla^2\rho$ are the gradient and Hessian of ρ, respectively. To do this, one usually needs to replace $\nabla\rho$ and $\nabla^2\rho$ by approximations based on difference ratios. The coefficient α_k is a step length chosen in such a way to ensure fast convergence to the minimum of ρ. This process is continued until the norm of $\nabla\rho(\theta_k)$ is less than some small given $\epsilon > 0$.

The part of Newton's method which requires the most computational power is the computation of the inverse of the Hessian. Quasi-Newton methods provide an attractive alternative in that they do not require the computation of this inverse Hessian at each step. Instead, they use an approximation which is updated after each step to take into account the additional knowledge gained during the step. The most effective Quasi-Newton updating formula was proposed by Broyden, Fletcher, Goldfarb, and Shanno, and is known as the BFGS method. This is the default optimization method in many software programs, for example, in Scilab [**38**].

The BFGS and Quasi-Newton methods have been well-studied by mathematicians, and their theoretical properties are well-established. As long as the step lengths and the Hessian approximations B_k produced satisfy certain conditions[1], they are known to converge quickly to the minimum. However, if the initial guess chosen is too far from the minimum, these methods may either diverge, or converge to a local minimum. Also, as these are deterministic methods, they only produce a point estimate of the coefficients.

3.2.2. Collocation methods. In this type of method, one estimates the coefficients θ of (3.2) by first using the known data to construct an approximation $\hat{y}^i(t)$ of each component $y^i(t)$ of the solution of (3.2) in terms of a basis function expansion

$$\hat{y}^i(t) = \sum_{j=1}^{J_i} c_{ij}\phi_{ij}(t),$$

where c_{ij} are real numbers and ϕ_{ij} are usually smooth piecewise-polynomial real functions known as *splines*. The number J_i of such functions used for the expansion depends on the amount of variation in $\bar{y}^i$: the more critical points the solution has, the larger J_i must be [**33**]. Then, to ensure fidelity with the equations of the differential system, this approximation and the corresponding approximate derivative are substituted in the differential equation (3.2). The coefficients which minimize the distance between $\hat{y}'$ and $g(\hat{y};\theta)$ using some norm are our resulting estimates.

This type of method has been available since 1982, in which Varah [**45**] used cubic splines and a least squares criterion to estimate the coefficients of a differential system. More recently however, Ramsay et. al. in [**33**] improved this method by changing the criterion to be optimized into a penalized least squares criterion:

$$J = \sum_i \left\{ w_i \|\bar{y}^i(t) - \hat{y}^i(T)\|^2 + PEN_i(\hat{y}) \right\}$$

[1]In particular, if there exists a constant M such that $\|B_k\|\|B_k^{-1}\| \leq M$ for all k where B_k is positive definite for all k, and the step lengths α_k satisfy sufficient decrease and curvature conditions such as the Wolfe conditions. (see [**28**], p. 45 for more details)

In the above equation, the w_i's are weights assigned so that the normalized error sum of squares are of roughly comparable sizes, and $PEN_m(\hat{y})$ measures the extent to which $\hat{y}^i$ satisfies the ODE system. For example, one choice of $PEN_i(\hat{y})$ could be

$$PEN_i(\hat{y}) = \int \left(\frac{d\hat{y^i}(t)}{dt} - g^i(\hat{y};\theta) \right)^2 dt,$$

where the integration is over an interval which contains the times of measurement T. Ramsay showed through numerical experiments that this method provides nearly unbiased estimates for the coefficients when simulating a FitzHugh-Nagumo model with Gaussian error.

Thus, by increasing the number of unknown coefficients from m to $m + \sum_i J_i$, the method allows us to handle problems with a lot of peaks and valleys. The spline basis expansion $\hat{y}^i$ provides us with the flexibility to capture more complex behavior between two data points, instead of just fitting a smooth curve between them. However, as the process usually involves minimizing a least-squares based criterion using, for example, a quasi-Newton method, it inherits the same disadvantages as in previous sections.

3.3. The rejection sampling algorithm

In this section, we describe a method that provides an alternative way to estimate θ. Instead of computing just a point estimate for θ, we wish to obtain the best possible *distribution* for the coefficients that fit the known data instead. The main idea is to produce a large sample of possible coefficients from some proposal distribution π_0, to compare the resulting trajectories with the known data, and to keep only those coefficients which give trajectories "close enough" to the data.

Here, we shall focus on the simplest way to do this, which is known as the rejection sampling (RS) method. The idea for this method was introduced as early as 1984 in a paper by Rubin [**37**] but was generalized by Pritchard et. al. [**32**] in 1999 in the context of population genetics. To estimate the coefficients of a system of differential equations, the method proceeds as follows: we begin by generating a sample $\{\theta_i^*\}_{i=1,2,...,N}$ of possible values of the coefficients from the prior distribution $\pi_0(\theta)$. We assume that this distribution has support S_0 which is usually compact. For each element θ^* of the sample, we compute the solution $y(t;\theta^*)$ of the differential equation (3.2) and keep its values $y(T;\theta)$ at times $T = \{t_0, t_1, ..., t_k\}$. Then, for a convenient measure $\rho(\theta^*)$ of the distance between the trajectory corresponding to the value θ^* of the coefficient and the known data $\bar{y}(T)$, if $\rho(\theta^*) < \epsilon$, where ϵ is a specified threshold constant, we

keep θ^*, and consider it as part of the sample from our approximate posterior; otherwise, we disregard this value. This process can then be repeated N times, or until we have some chosen number, say n, accepted values of θ. One can then construct the histogram of a sample of values from this distribution, or compute the summary statistics from the sample. Here, the measure

$$\rho(\theta^*) = \rho(\{y(t_i;\theta^*), i = 1, ..., k\}, \{y_i, i = 1, ..., k\}). \tag{3.3}$$

can be, for example, the sum of squares of the Euclidean distance

$$\rho(\theta^*) = \sum_{i=1}^{k} \|y(t_i;\theta^*) - y_i\|^2$$

or other ones. We shall call the sample produced in this manner a *rejection sample.*

DEFINITION 3.2. *Let $\theta_1, \theta_2, ..., \theta_N$ be an i.i.d. sample from π_0 and let n be the largest ν for which there exists a sequence of integers $(i_k)_{k=1,2,...,\nu}$ such that $1 \le i_1 < i_2 < ... < i_\nu \le N$ and $\rho(\theta_{i_k}) < \epsilon$ for all k. We call $\theta_{i_1}, \theta_{i_2}, ..., \theta_{i_n}$ a* ***rejection sample*** *of size n. The* ***acceptance rate*** *of this sample is given by $\tau_{RS} = n/N$.*

For simplicity, whenever no confusion arises, we shall denote a rejection sample of size n as $\theta_1, \theta_2, ..., \theta_n$ instead of $\theta_{i_1}, \theta_{i_2}, ..., \theta_{i_n}$.

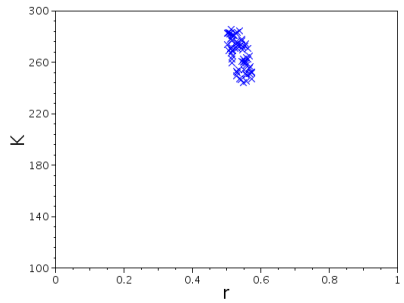
REMARK 3.3. By construction, for a given ϵ, the rejection sample $\theta_1, \theta_2, ..., \theta_n$ is drawn from the distribution

$$\pi_\epsilon(\theta|\bar{y}) = \frac{\pi_0(\theta)\mathbb{1}_{A_\epsilon}(\theta)}{\int_{A_\epsilon} \pi_0(\theta)d\theta}, \tag{3.4}$$

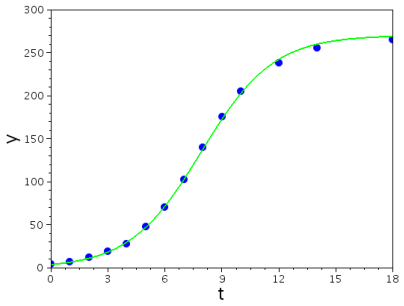
where A_ϵ is the *acceptance region*

$$A_\epsilon = \{\theta \in S_0 | \rho(\theta) < \epsilon\}. \tag{3.5}$$

EXAMPLE 3.4. Before studying the properties of rejection samples, let us first see how the method works on the logistic equation which we studied in Example 3.1. Here, $\theta = (r, K)$, and we take the uniform distribution over $[0, 1] \times [100, 300]$ as our prior distribution π_0 for θ. The value of ϵ is chosen as 1300, which represents an average error of 10 units for each of the 13 time points which are allowed to vary. The method is run for $N = 5000$ iterations, and for this particular run, we ended up with 65 accepted coefficients. The mean of the accepted values is approximately $(0.5349, 267.61)$ while the θ_i that gives the smallest distance is about $(0.5351, 265.94)$. The results are given in the following figure. One can see that the fit is better compared to that of the crude least squares method in Example 3.1. Notice that even with a relatively large value of ϵ, the resulting sample is distributed in a small elliptical region, which is nothing else than A_ϵ.



(A) Plot of the resulting sample in the rectangle which is the support S_0 of the prior



(B) Graph of the logistic curve with "minimum distance" coefficients

FIGURE 3.2. Results for logistic model using rejection sampling algorithm

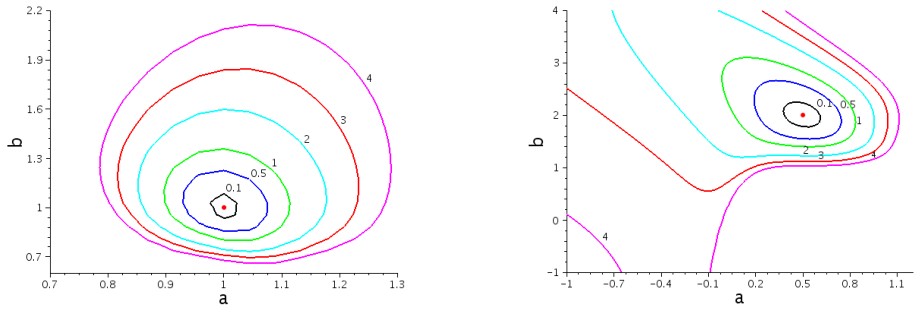
3.4. An analysis of the rejection sample

In this section, we shall assume that the observed data are the actual values of the trajectory $y(T;\theta)$ where the value of the coefficient is θ_0 and the time points are given by $T = \{t_0, t_1, ..., t_k\}$. In this case, we shall call θ_0 the "true" value of the coefficient. While this may not be consistent with actual data, this assumption is useful to study separately the accuracy of the method and its robustness with respect to noisy data. We shall look at the application of the method to a more "realistic" problem with noise in section 3.6.

3.4.1. The acceptance region. In this section, we will study the size and shape of the acceptance region A_ϵ, defined by (3.5). We shall mainly focus on just two-variable coefficient estimation problems. Such a simple model will allow us to understand the properties more easily, as this will make the graph of the acceptance region A_ϵ easier to visualize.

The size and shape of the region A_ϵ are clearly dependent on the value of ϵ and the differential system through the function $\rho(\theta)$. Obviously, as soon as ϵ is larger than the maximum of $\rho(\theta)$ for $\theta \in S_0$, the acceptance region will be the entire support S_0. Otherwise, it will be helpful to look at the contour map for $\rho(\theta)$ to understand the shape of the acceptance region better. Let us first take a look at the contour map of $\rho(\theta)$ for some examples. Figure 3.3 gives the contour maps for $\rho(\theta)$ for the two-coefficient Lotka-Volterra model introduced in Section 1.4.3, for two different values of θ_0 are shown.

These contour maps were obtained in scilab by evaluating $\rho(\theta)$ for a 70 by 70 grid of evenly-spaced values in $[-1, 2] \times [-1, 4]$. The `contour2d` command is called to produce only the contours for $\rho = 0.1, 0.5, 1, 2, 3, 4$ for clarity. The contour line for $\epsilon = 1$ is highlighted. One can see that for ϵ small enough, the



(A) True coefficient value: $\theta_0 = (a, b) = (1, 1)$, initial point $(0.5, 1.5)$

(B) True coefficient value: $\theta_0 = (a, b) = (0.5, 2)$, initial point $(1, 0.4)$

FIGURE 3.3. Contour maps for the two-coefficient Lotka-Volterra model. In both cases, we assume that the "discrete" trajectory is defined for $T = \{0, 1, 2, ..., 7\}$, ρ is the sum of squared differences, and we draw only those values of (a, b) within the region $[-1, 2] \times [-1, 4]$. The point in red corresponds to θ_0.

boundary of A_ϵ looks like an ellipse. It turns out that this is to be expected, even for higher-dimensional cases. As we have assumed in this section that θ_0 is a "true" value of the parameter, thus, by definition $\rho(\theta_0) = 0$ and θ_0 is the minimum of ρ. We can then make the following remark:

REMARK 3.5. If the prior distribution is chosen such that θ_0 belongs to its support, then whenever the system is of class C^2, one can expect that the region $\rho < \epsilon$ has the shape of an ellipsoid around θ_0 as soon as ϵ is small enough.

Indeed, $y(T; \theta)$ is also of class C^2, and consequently, $\rho(\theta)$ as well. Thus, we can take the second-order Taylor expansion of ρ at its minimum θ_0:

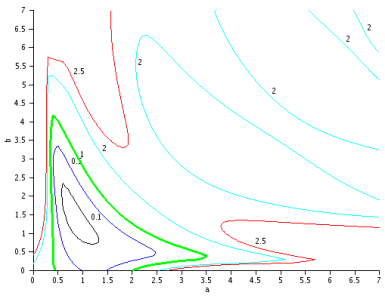
$$\rho(\theta) = \rho(\theta_0) + (\nabla\rho(\theta_0))^T \Delta\theta + \frac{1}{2}\Delta\theta^T H(\rho(\theta_0))\Delta\theta + O(\|\Delta\theta\|^3)$$

where $\Delta\theta = \theta - \theta_0$, ∇ and H are the gradient and Hessian of ρ respectively. Since θ_0 is a minimum value of ρ, $\nabla\rho(\theta_0)$ is the zero vector. Thus, if we fix ϵ and neglect the higher-order terms, the set of possible θ for which $\rho(\theta) < \epsilon$, satisfies $\frac{1}{2}\Delta\theta^T H(\rho(T, \theta_0))\Delta\theta < \epsilon$, or equivalently,

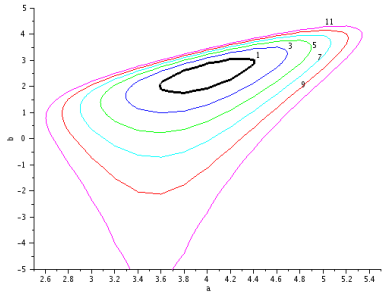
$$\frac{1}{2}\Delta\theta^T H(\rho(\theta_0))\Delta\theta < \epsilon, \tag{3.6}$$

which has the interior of an ellipsoid around θ_0 as a graph because H is positive definite.

However, we note that the value of ϵ that is "small enough" varies depending on the differential system. This can be seen in Figure 3.4a, where we have the contour map for ρ for the harmonic oscillator (see equation 1.20). On the other



(A) Contour map for the harmonic oscillator model, where the "true" coefficients $a = 0.8, b = 1.5$, initial point $(0.3, 0.2)$, times $T = \{0, 0.5, 1, ..., 3\}$, and limited to the range $[0, 7] \times [0, 7]$



(B) Contour map for the competing species model

FIGURE 3.4. Contour maps

hand, Figure 3.4b is for a non-oscillatory differential system, the competing species model

$$\begin{cases} \dfrac{dy_1}{dt} &= ay_1 - y_1^2 - 0.5y_1y_2 \\ \dfrac{dy_2}{dt} &= by_2 - 0.5y_2^2 - 1.5y_1y_2 \end{cases} \tag{3.7}$$

where the reference trajectory has "true" coefficients $a = 4, b = 2.5$, initial point $(0.5, 3)$, times $T = \{0, 0.5, 1, ..., 3\}$ and range $[0, 10] \times [-5, 5]$. In both examples, ρ is the sum of squared differences. As before, only a few contour lines are included in both figures for clarity. From these, notice that for the contour to become approximately an ellipse, ϵ must be around 0.1 in the case of our harmonic oscillator. However, in our competing species example, A_ϵ is still close to an ellipse even when $\epsilon = 7$.

In contrast, for larger values of ϵ, there is no reason that the acceptance region will be an ellipse. In fact, it may even be neither centered at θ_0, nor consist of a single region. To show how the shape of ρ may change dramatically, an example is given in Figure 3.5, where we have separated the contour lines of ρ for the harmonic oscillator in Example 3.4a.

Furthermore, we encounter a very interesting situation when $\epsilon = 2$. Here, the acceptance region consists of several closed regions. This occurs when $\rho(\theta)$ has several local minima. As this existence of several minima is important to understand not only in the case of the rejection sampling method but in all such methods which use the minimisation of a criterion ρ, let us examine now more closely how and when this phenomenon occurs. To do this, we look at an even

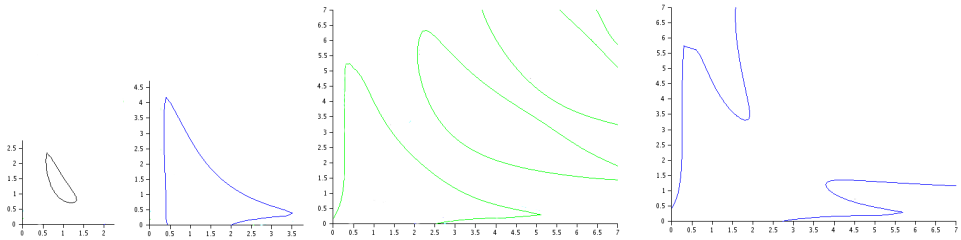


FIGURE 3.5. Shape of acceptance regions for the harmonic oscillator as ϵ increases

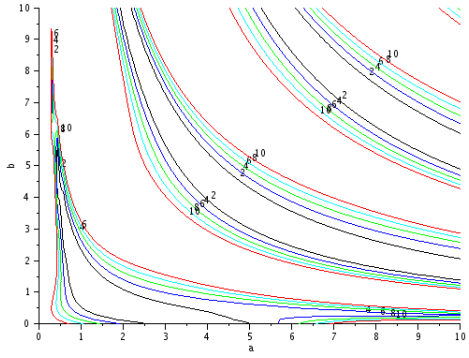
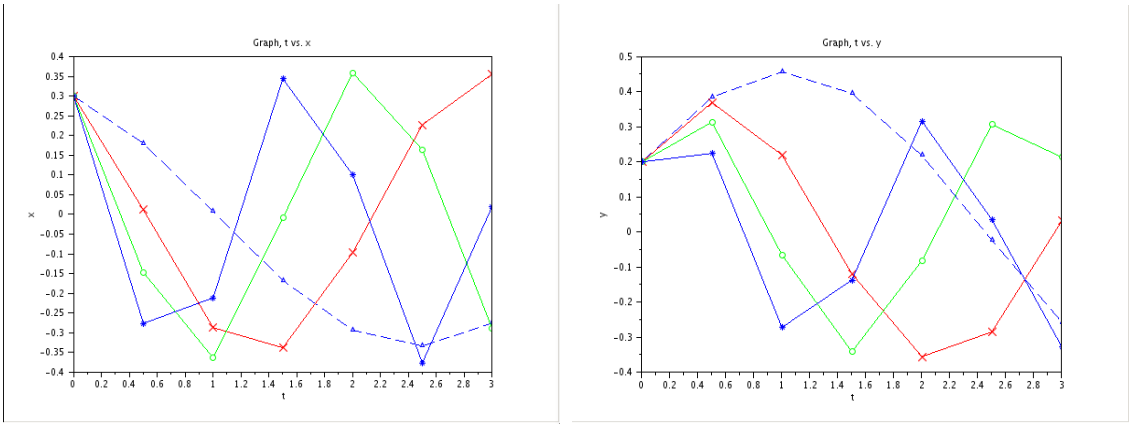


FIGURE 3.6. Contour map for the harmonic oscillator with only two points for T

simpler example, which is the harmonic oscillator with $a = 1, b = 1.5$ and initial point $(2, 0.5)$, but now with just two points in the reference trajectory, at $t = 0$ and $t = 2$. We still use the sum of the squared differences as our metric, with only one term in this case. The resulting contour map over $S = [0, 10] \times [0, 10]$ is shown in Figure 3.6.

We see that within S, $\rho(\theta)$ achieves a (global) minimum three times – aside from $(a, b) = (1, 1.5)$ which we expected to obtain, we also have two other global minima, which occur at around $(1 + p\pi/\sqrt{1.5}, 1.5(1 + p\pi/\sqrt{1.5}))$, for $p = 1, 2$. To see this, note first from (1.21) that while the period of the entire solution is equal to $2\pi/\sqrt{ab}$, the amplitude of $y(t)$ varies, and is equal to $\sqrt{\frac{b}{a}}$. But if we fix $b/a = 1.5$ which is the same as the true value $(1, 1.5)$, the solution at $t = 2$ reduces to

$$\begin{cases} x(a) &= K_1 \cos 2\sqrt{1.5}a - K_2 \sin 2\sqrt{1.5}a \\ y(a) &= K_1\sqrt{1.5} \sin 2\sqrt{1.5}a + K_2\sqrt{1.5} \cos 2\sqrt{1.5}a \end{cases} \tag{3.8}$$



(A) (B)

FIGURE 3.7. Graphs of $x(t)$ and $y(t)$ for four different choices of (a, b). In order along the line segment from $(0.8, 1.5)$ to $(4, 3)$, we have $\wedge, \mathrm{x}, \mathrm{o}, *$. Due to the periodic nature of the trajectories, the distance of each point from the data is not monotonically increasing as you go farther from the true coefficients.

This has a period of $\pi/\sqrt{1.5}$. Thus, $\rho = 0$ for $a = 1 + p\pi/\sqrt{1.5}$ and $b = 1.5a$ for any integer p. Here, we can achieve the minimum value for ρ with many different ordered pairs (a, b). This is related to the problem of *identifiability* of any optimization method. For all our succeeding discussions after this section, we will assume that the coefficients θ are uniquely identifiable.

When there are more than two points in our data that need to be satisfied, it becomes more difficult to understand precisely the mechanism that results in multiple local minima. It seems, however, that having multiple sets of coefficients giving the exact global minimum becomes rarer. In the case of $\epsilon = 2$ in Figure 3.4a, we see that while there are multiple local minima, there is only one global minimum, and this is at $(0.8, 1.5)$. To help understand what exactly happens with ρ, consider the line segment in the ab-plane joining $(1, 1.5)$ and the point $(4, 3)$, which is a point close to one of the local minima. As you go along this line segment, ρ increases initially, and then decreases until you reach $(4, 3)$. To see why this is the case, we divide the line segment into 3 equal parts. Then, we will plot the trajectories for $(0.8, 1.5)$, $(1.87, 2)$, $(2.93, 2.5)$, and $(4, 3)$. Using these points as our coefficients, we then plot in the $t - x$ and $t - y$ planes the resulting seven points in the discrete trajectory.

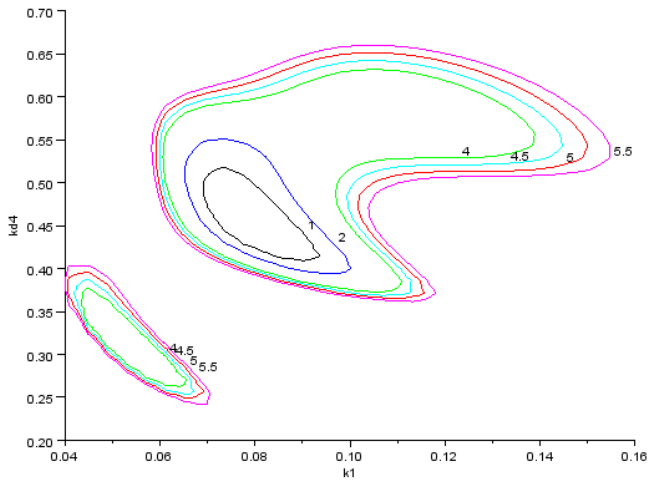
We see in Figure 3.7 that due to the periodic nature of the trajectories, the distance from the data at each time point is not monotonically increasing as you go farther away from $(0.8, 1.5)$. While the points obtained using $(a, b) = (2.93, 2.5)$ are farther than those from $(a, b) = (1.87, 2)$, those at $(a, b) = (4, 3)$ are generally *closer* than those from $(2.93, 2.5)$.

EXAMPLE 3.6. The problem of multiple local maxima can also occur in more complicated systems with periodic behavior. Consider the system used to model circadian cycles as defined in (1.24). Suppose all the coefficients are known except for k_1 and kd_4, where these coefficients have values based on the "common coefficients" in page 85 of [**3**]. Substituting these reduces the differential system to the following:

$$\text{(3.9)} \qquad \begin{cases} \dfrac{dp_1}{dt} &= 2 \cdot \dfrac{0.4^{15}}{0.4^{15} + c_2^{15}} - 0.08 p_1 p_2 + k_4 c_1 - k d_1 p_1 \\ \dfrac{dp_2}{dt} &= 2.2 \cdot \dfrac{0.4^{15}}{0.4^{15} + c_2^{15}} - 0.08 p_1 p_2 + 0.06 c_1 - 0.05 p_2 \\ \dfrac{dc_1}{dt} &= 0.08 p_1 p_2 - 0.06 c_1 - k_1 c_1 + 0.06 c_2 - 0.05 c_1 \\ \dfrac{dc_2}{dt} &= k_1 c_1 - (0.06 + k d_4) c_2 \end{cases}$$

The known data is assumed to be the actual trajectory when $k = 0.08$ and $kd_4 = 0.45$ when $T = \{0, 11, 12, 50, 51, 52, 53, 54\}$. We see in Figure 3.8 of the presence of two disjoint regions in A_ϵ in this case.

FIGURE 3.8. The contour plot for $\rho(\theta)$, where the initial point is chosen to be $(1.5, 2, 1, 0.5)$.



Based on the two previous examples, it seems reasonable therefore to claim the following:

CONJECTURE 3.7. *Suppose that $\rho(\theta)$ is the metric to be minimized when estimating the coefficients θ in a system of differential equations $y = g(y; \theta)$,*

where ρ is as defined in (3.3)*. Then ρ has several local minima if the solution $y(t;\theta)$ is periodic, and where the period is a function of θ.*

3.4.2. Point estimation. Using the accepted coefficients produced by our method, there are two natural ways to construct an estimate for θ_0: one can either take the average $\widehat{\theta}_{ave}$ of the entire rejection sample or take $\widehat{\theta}_{min}$, which is the one which produces the minimum distance from the given data using the metric ρ. We shall see in this section that for the case where θ_0 is the true value of the coefficients, the one which produces the minimum distance is clearly the better choice.

As previously mentioned, the two summary statistics from the rejection sample $\theta_1, \theta_2, ..., \theta_n$ will be the sample average, denoted by $\widehat{\theta}_{ave}$ and the minimum distance, denoted $\widehat{\theta}_{min}$. These are defined more precisely as follows:

$$\widehat{\theta}_{ave} = \frac{1}{n}\sum_{i=1}^{n}\theta_i,$$

where the sum is taken component-wise, and

$$\widehat{\theta}_{min} = \underset{\theta\in\{\theta_1,\theta_2,...,\theta_n\}}{\text{ArgMin}}\ \rho(\theta).$$

By the Law of Large Numbers, we know that $\hat{\theta}_{ave}$ converges almost surely to the expectation of θ with distribution $\pi_0(\theta|\rho(\theta) < \epsilon)$. This need not be equal to θ_0, especially for large ϵ. This is true even for a uniform prior, as the region may not be centered on θ_0, as we have seen in the previous section. In contrast, $\hat{\theta}_{min}$ has a very nice property as an estimator for θ_0, as we now show.

PROPOSITION 3.8. *Let θ_1, θ_2,..., θ_n be a rejection sample for threshold value $\epsilon > 0$ to estimate the coefficients θ in a differential equation $y' = g(y;\theta)$, and let*

$$\hat{\rho}_n(\theta) = min\{\rho(\theta_1), \rho(\theta_2), ..., \rho(\theta_n)\}.$$

Assume also that the prior distribution π_0 for the rejection sample is absolutely continuous. If $\theta_0 = \underset{\theta\in S_0}{ArgMin}\rho(\theta)$, then $\hat{\rho}_n(\theta) \xrightarrow{\mathcal{P}} \rho(\theta_0)$ as the sample size n tends to ∞.

Proof. Let ρ_i denote the distance $\rho(\theta_i)$. For any $K > 0$, first note that, as $\theta_1, ..., \theta_n$ is a rejection sample and thus i.i.d., we have

$$\mathbb{P}(\hat{\rho}_n(\theta) \le K) = 1 - [\mathbb{P}(\rho_1 \ge K)]^n \tag{3.10}$$

Indeed:

$$\begin{aligned}\mathbb{P}(\min(\rho_1, \rho_2, ..., \rho_n) \le K) &= 1 - \mathbb{P}(\min(\rho_1, \rho_2, ..., \rho_n) \ge K)\\ &= 1 - \mathbb{P}(\rho_1 \ge K, \rho_2 \ge K, ..., \rho_n \ge K)\\ &= 1 - [\mathbb{P}(\rho_1 \ge K)]^n\end{aligned}$$

Thus, for any $\alpha > 0, n > 0$,

$$\begin{aligned}\{|\hat{\rho}_n(\theta) - \rho(\theta_0)| < \alpha\} &= \{\rho(\theta_0) - \alpha < \hat{\rho}_n(\theta) < \rho(\theta_0) + \alpha\} \\ &= \{\hat{\rho}_n < \rho(\theta_0) + \alpha\} \setminus \{\hat{\rho}_n \leq \rho(\theta_0) - \alpha\}\end{aligned}$$

As $\{\hat{\rho}_n < \rho(\theta_0) - \alpha\} \subset \{\hat{\rho}_n \leq \rho(\theta_0) + \alpha\}$, we have

$$\begin{aligned}\mathbb{P}(|\hat{\rho}_n(\theta) - \rho(\theta_0)| < \alpha) &= \mathbb{P}(\hat{\rho}_n < \rho(\theta_0) + \alpha) - \mathbb{P}(\hat{\rho}_n \leq \rho(\theta_0) - \alpha) \\ &= [\mathbb{P}(\rho_1 > \rho(\theta_0) - \alpha)]^n - [\mathbb{P}(\rho_1 \geq \rho(\theta_0) + \alpha)]^n, \qquad (3.11)\end{aligned}$$

where the last line follows from (3.10). The limit of the first term of (3.11) as $n \to \infty$ is 1 since by definition, $\rho(\theta_0)$ is the minimum. On the other hand, the limit of the second term is 0 since $\mathbb{P}(\rho_1 \geq \rho(\theta_0) + \alpha) < 1$ from the absolute continuity of $\pi_0(\theta)$. Thus, $\lim_{n\to\infty} \mathbb{P}(|\hat{\rho}_n(\theta) - \rho(\theta_0)| < \alpha) = 1$, and so $\hat{\rho}_n(\theta)$ converges in probability to $\rho(\theta_0)$. □

As an illustration of Proposition 3.8, let us consider a simple example that shows that increasing the sample size does not improve $\widehat{\theta}_{ave}$ but significantly improves $\widehat{\theta}_{min}$. Tables 3.2 and 3.1 show how the resulting estimates $\widehat{\theta}_{ave}$ and $\widehat{\theta}_{min}$ vary when estimating the coefficients in the repressilator model (see equation 1.23) as the size of the sample increases while holding everything else constant. Here, we assume that the true values of the coefficients are $(\alpha, \alpha_0, \gamma, \beta) = (1000, 1, 2, 5)$ and the $\epsilon = 2000$. Looking at the results in Table 3.1, we can see that $\hat{\rho_N}(\hat{\theta}_{ave})$ does not decrease for the estimates using $\widehat{\theta}_{ave}$ as N increases. However, the values of $\hat{\alpha_0}_{ave}, \hat{\alpha}_{ave}, \hat{\gamma}_{ave}$ and $\hat{\beta}_{ave}$ are consistently around 1010, 1.30, 2.05, and 5.8, respectively, which we can assume to be close to the actual expectation of the distribution $\pi(\theta|\rho(\theta) < 2000)$. In contrast, the results in Table 3.2 show that while the values of $\hat{\rho}_N(\hat{\theta}_{min})$ may not always decrease monotonically as N increases (due to the random nature of the sample), there is still a clear trend of decrease for ρ as $N \to \infty$. This exactly illustrates Proposition 3.8.

TABLE 3.1. Results of $\widehat{\theta}_{ave} = (\hat{\alpha}_{ave}, \hat{\alpha_0}_{ave}, \hat{\gamma}_{ave}, \hat{\beta}_{ave})$ for $\epsilon = 2000$. True value: $\alpha = 1000, \alpha_0 = 1, n = 2, \beta = 5$, prior distribution: uniform over $[800, 1200] \times [0, 4] \times [0, 7] \times [0, 10]$

N	$\hat{\alpha}_{ave}$	$\hat{\alpha_0}_{ave}$	$\hat{\gamma}_{ave}$	$\hat{\beta}_{ave}$	$\hat{\rho}_N(\hat{\theta}_{ave})$
500	1004.3	1.1800	2.0730	6.1476	265.03
1000	1008.0	1.3508	2.0534	5.6957	337.78
2000	1005.1	1.3364	2.0394	5.7085	412.41
4000	1022.3	1.2380	2.0354	5.8069	342.04
8000	1010.5	1.3224	2.0521	5.8860	403.34
16000	1011.0	1.2886	2.0409	5.8135	390.4

TABLE 3.2. Results of $\widehat{\theta}_{min} = (\hat{\alpha}_{min}, \hat{\alpha_0}_{min}, \hat{\gamma}_{min}, \hat{\beta}_{min})$ for $\epsilon = 2000$. True value: $\alpha = 1000, \alpha_0 = 1, \gamma = 2, \beta = 5$, prior distribution: uniform over $[800, 1200] \times [0, 4] \times [0, 7] \times [0, 10]$

N	$\hat{\alpha}_{min}$	$\hat{\alpha_0}_{min}$	$\hat{\gamma}_{min}$	$\hat{\beta}_{min}$	$\hat{\rho}_N(\hat{\theta}_{min})$
500	1158.6	0.2709	1.7097	3.8584	352.06
1000	1142.2	0.7723	1.9288	4.8699	138.67
2000	914.4	0.2616	1.801	4.0841	136.64
4000	819.77	0.9172	2.0406	5.143	37.59
8000	845.32	0.9282	2.0363	5.0312	38.71
16000	859.59	0.9781	2.0452	5.1839	26.57

REMARK 3.9. Given that we have a fixed number of accepted elements in the sample in each case, the point estimates obtained using both the average and the minimum distance get more precise as ϵ decreases. This is not surprising because as ϵ decreases, the algorithm becomes more selective. Also, since for small ϵ, the graph of A_ϵ becomes that of an ellipsoid, we can deduce that as $\epsilon \to 0$, the acceptance region converges to that of a point ellipsoid centered at the minimum θ_0. Thus, $\pi_\epsilon(\theta|\bar{y}(T))$ will tend to the Dirac distribution in this unknown value.

However, in practice, this will not be the case since as ϵ decreases, the acceptance rate also decreases, as we will state more formally in the next section.

In all our results in this section, we have assumed that θ_0 is a "true" value of the coefficients, and so the known data $\bar{y}$ is generated perfectly as a solution of the differential system when $\theta = \theta_0$. In section 3.5.3 below, we will revisit the question of which is the best estimate for θ. We will see that we can do even better by combining the minimum and an average.

3.4.3. The acceptance rate. An important factor in the success of the rejection approach is the size of the rejection sample. As we have seen in the previous section, the larger the resulting rejection sample, the higher the chances of obtaining a more accurate estimate for θ using the minimum distance estimator. In the discussion that follows, we will see how the acceptance rate is affected by the sample size N, the maximum threshold ϵ, and the number of coefficients m.

Let $\theta_1, \theta_2, ..., \theta_N$ be an i.i.d. sample from π_0. Define the corresponding random variables $I_i = \mathbb{1}_{\{\theta_i \in A_\epsilon\}}, i = 1, 2, ..., N$. Then, $I_1, I_2, ..., I_N$ are i.i.d. Bernoulli random variables with success probability

$$\tau = \mathbb{P}(\theta \in A_\epsilon) = \int_{A_\epsilon} \pi_0(\theta) d\theta. \tag{3.12}$$

Thus, by the Law of Large Numbers, as $N \to \infty$, the rejection sampler acceptance rate τ_{RS} converges in probability to τ.

REMARK 3.10. *When the prior distribution π_0 is uniform,* (3.12) *implies that the acceptance rate is simply the ratio of the volumes of A_ϵ and S_0.*

Furthermore, by the Central Limit Theorem,

$$\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^{N} I_i - \tau\right) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \tau(1-\tau)).$$

This means that provided N is large enough, a 95% confidence interval for the acceptance rate τ_{RS} is approximately given by $\tau \pm 1.64\sqrt{\frac{\tau(1-\tau)}{N}}$. Since the maximum of $\tau(1-\tau)$ occurs when $\tau = 0.5$, a conservative estimate for the length of the confidence interval is $1.64/\sqrt{N}$. In general, τ cannot be computed directly, and we will use its estimator τ_{RS} instead. For large N, this confidence interval implies that we can actually expect the acceptance rate to be more or less the same, and to be on a narrow range around τ.

EXAMPLE 3.11. Consider the acceptance rate when estimating the coefficients of a harmonic oscillator model as the size N of the rejection sample increases. The plot of N in relation to the acceptance rate is given in Figure 3.9. One can see that for small values of N, the amplitudes of the oscillations are larger. However, as N increases, we can see that the acceptance rate remains in a small region around 0.002.

Although the sample size N may not play a large role on the acceptance rate of our sample, it is clear that this is not the same with the maximum threshold ϵ. In fact, for any sample $\theta_1, \theta_2, ..., \theta_N$, the asymptotic acceptance rate τ is a nondecreasing function of ϵ. To see why this is so, denote by τ_ϵ^N the acceptance rate associated with the threshold ϵ and let $\epsilon_1 < \epsilon_2$. Then $A_{\epsilon_1} \subset A_{\epsilon_2}$, and so $\tau_{\epsilon_1}^N = \mathbb{P}(\theta \in A_{\epsilon_1}) \leq \mathbb{P}(\theta \in A_{\epsilon_2}) = \tau_{\epsilon_2}^N$, thus, this inequality also holds for the asymptotic acceptance rate.

When the prior distribution is uniform, we can say even more about τ. In particular, the rate of increase of the acceptance rate is of the order $l/2$, where l is the dimension of the coefficient space, as we shall now show.

PROPOSITION 3.12. *Suppose that the rejection method is used to estimate θ in the differential system $y' = g(y;\theta)$, where the known data $\bar{y}$ consists of the actual points for $y(t;\theta)$ when $\theta = \theta_0$. Assume also that θ is l-dimensional, and that the prior distribution is uniform on its support. Then, the acceptance rate τ_{RS} satisfies*

$$\lim_{N\to\infty} \tau_{RS} = \frac{(2\pi\epsilon)^{l/2}}{\Gamma(\frac{l}{2}+1)\sqrt{\prod_{i=1}^{l}\lambda_i}} \tag{3.13}$$

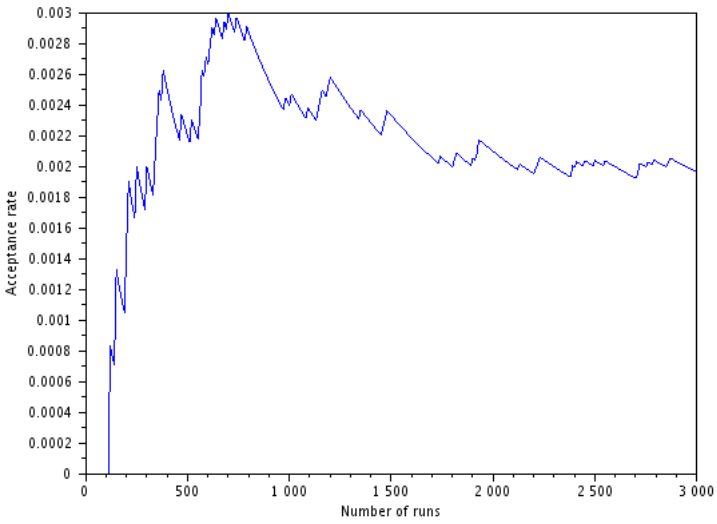


FIGURE 3.9. Acceptance rate for a harmonic oscillator as a function of N. We set the "true" coefficients to be $(0.8, 1.5)$, the initial point as $(0.3, 0.2)$, $T = \{0, 0.5, 1, ..., 3\}$, and $\epsilon = 0.1$.

where $\lambda_1, \lambda_2, ..., \lambda_l$ *are the eigenvalues of the Hessian matrix of* $\rho(\theta_0)$ *and* Γ *is the gamma function*

$$\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} du.$$

Proof. Since θ_0 is a true value of the coefficients and ϵ is small enough, by (3.6), the acceptance region reduces to the ellipsoid

$$\Delta\theta^T H(\rho(\theta_0))\Delta\theta < 2\epsilon, \tag{3.14}$$

where $H(\rho)$ is the Hessian of ρ. Since H is a Hessian matrix, it is symmetric, and so we can diagonalize it with an orthogonal matrix Q:

$$H = Q\Lambda Q^T,$$

where Λ is the diagonal matrix of eigenvalues of H. Substituting in (3.14), we get

$$\begin{aligned} 2\epsilon &> \Delta\theta^T Q\Lambda Q^T \Delta\theta \\ \epsilon &> v^T \left(\frac{1}{2}\Lambda\right) v \end{aligned} \tag{3.15}$$

where $v = Q^T\Delta\theta$. Since $\det\Lambda$ is simply the product of the eigenvalues $\lambda_1, \lambda_2, ..., \lambda_l$ of H, by Proposition 1.29, we can conclude that the volume of the ellipsoid is

$$\frac{(\pi\epsilon)^{l/2}}{\Gamma(\frac{l}{2}+1)} \cdot \left(\det\frac{1}{2}\Lambda\right)^{-1/2} = \frac{(2\pi\epsilon)^{l/2}}{\Gamma(\frac{l}{2}+1)\sqrt{\prod_{i=1}^{l}\lambda_i}}, \tag{3.16}$$

which is as claimed. As the asymptotic acceptance rate is simply the ratio of volumes when the prior distribution is uniform, the proof is complete. □

REMARK 3.13. *Letting* $v = (x_1, x_2, ..., x_l)^T$ *in* (3.15)*, we can convert the ellipsoid* (3.6) *to the unrotated ellipsoid*

$$\frac{1}{2}(\lambda_1 x_1^2 + \lambda_2 x_2^2 + ... + \lambda_l x_l^2) = \epsilon,$$

where $\lambda_1, \lambda_2, ..., \lambda_l$ *are the eigenvalues of* H*. This tells us that the lengths of the semi-principal axes of* (3.6) *are*

$$\sqrt{\frac{2\epsilon}{\lambda_1}}, \sqrt{\frac{2\epsilon}{\lambda_2}}, ..., \sqrt{\frac{2\epsilon}{\lambda_l}}.$$

EXAMPLE 3.14. We return to the logistic model in Example 3.4, where we found that the best fit coefficients were approximately $r_0 = 0.5351$ and $K_0 = 265.94$. Recall that in that example, the prior distribution was chosen to be uniform on the rectangle $[0,1] \times [100,300]$. Assuming the initial point $(0,4)$, the objective function that we wish to minimize can be computed exactly as

$$\rho(r,K) = \sum_{i=0}^{13} \left(\frac{4Ke^{rt_i}}{K - 4 + 4e^{rt_i}} - \bar{y}_i \right),$$

where (t_i, y_i) are the 14 data points which were previously given. Assuming that $\theta_0 = (r_0, K_0)$, we shall use $\epsilon - \rho(\theta_0) \approx 1266.2$ on the right-hand side of (3.6). Using a symbolic computation program such as Maple, one can compute the Hessian of ρ at θ_0:

$$H = \begin{pmatrix} 2516643.593 & 2220.7277 \\ 2220.7277 & 6.264013 \end{pmatrix}, \tag{3.17}$$

which has eigenvalues $\lambda_1 = 4.3044$ and $\lambda_2 = 2516645.6$. This means that the semi-principal axes are $\sqrt{2532.4/4.3044} \approx 24.255$ and $\sqrt{2532.4/25616645.6} \approx 0.03172$. If we assume that the prior distribution π_0 is uniformly distributed on its support, the acceptance rate is thus $\tau = \pi(24.255)(0.03172)/200 \approx 0.01208$. Since the acceptance rate that we obtained from the sample before was $65/5000 = 1.3\%$ and the sample size was $N = 5000$, the 95% confidence interval for τ is thus

$$0.013 \pm 1.645\sqrt{\frac{(0.013)(0.987)}{5000}} = (0.01036, 0.01564),$$

which contains the asymptotic acceptance rate $\tau = 0.01208$.

Table 3.3 shows the 95% confidence interval for the asymptotic acceptance rate when estimating the coefficients of the two-coefficient Lotka-Volterra model. Here, we assumed that the true values of the coefficients are $a = 1$ and $b = 1$, initial point $(1, 0.5)$, $T = \{0, 1, ..., 7\}$, $N = 1000$, and a uniform prior distribution over the square $[0,3] \times [0,3]$. We see that as ϵ decreases, τ also decreases.

TABLE 3.3. Results for Lotka-Volterra with $a = 1, b = 1$ when ϵ varies

ϵ	95% confidence interval for τ (in percent)
3	$(5.14, 6.62)$
2	$(3.51, 4.77)$
1	$(1.20, 2.00)$
0.5	$(0.63, 1.25)$

Aside from the size of ϵ, there are other factors which affect the acceptance rate. Clearly, if θ is of large dimension, one will get a low acceptance rate. We illustrate this in the following example.

EXAMPLE 3.15. Consider the repressilator model for gene regulatory networks (1.23). We assume that the initial point is at $(0, 2, 0, 1, 0, 3)$ and that there is a true value of the coefficients, namely $(\alpha_0, \gamma, \beta, \alpha) = (1, 2, 5, 1000)$. Seven points were chosen to represent the trajectory, in particular at $t = 0, 1, 2, 3, 4, 5, 6$. The prior distribution is uniformly distributed on $[0, 2] \times [1.5, 2.5] \times [0, 10] \times [900, 1100]$. We produced a sample of size 10000, and kept only those which yield trajectories having a maximum distance of 60. The number of unknown coefficients is different in each experiment. As the acceptance rate varies on each experiment, we take the average of the acceptance rate when producing five separate rejection samples (which we shall call five *runs* later). The results are given in Table 3.4, along with the corresponding acceptance rate.

TABLE 3.4. Acceptance rate of the repressilator (1.23) when the number of unknown coefficients varies. The results shown are the average acceptance rate for five separate runs.

Coefficients fixed	Average acceptance rate (in percent)
None	0.404
α	0.474
α, α_0	1.372
α, α_0, γ	10.2

From Table 3.4, we see that as the dimension of the coefficient space increases, the corresponding acceptance rate decreases, where it is less than 1 percent for the case of 3 or more variables. This is even if the support of the prior distribution that we have chosen covers a region which is a small neighborhood of θ_0. This means that one will need a larger sample to obtain a relatively accurate estimate of θ. We will see in the next chapter several methods to alleviate this situation.

While it may seem logical that the acceptance rate τ varies directly as the size of the support of π_0, this is not true in general. As an example when this is not the case, consider again the repressilator model (1.23) with true

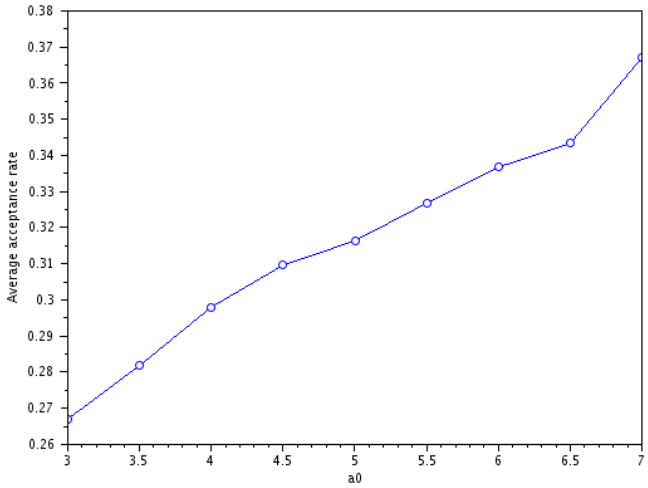


FIGURE 3.10. Acceptance rate of the repressilator model (1.23) as a function of the maximum α_0 in the prior

coefficients $(\alpha_0, \gamma, b, \alpha) = (1, 2, 5, 1000)$, initial point $(0, 2, 0, 1, 0, 3)$, time points $T = \{0, 1, 2, ..., 7\}$, and $\epsilon = 10000$. We compare the acceptance rates using a uniform prior, but this time, where the intervals for γ, b and α are $[0, 4]$, $[0, 10]$, and $[800, 1200]$, and α_0 is from the interval $[0, \alpha_0^+]$, where α_0^+ ranges from 3 to 7 in increments of 0.5. The average acceptance rate when five separate rejection samples (five runs) is given in Figure 3.10.

We can see that as α_0^+ increases, the acceptance rate actually also increases, which is completely counterintuitive. The reason for this apparent "paradox" is the shape of the acceptance region. Figure 3.11 shows the scattermatrix for the four coefficients in θ. We can see in the scatterplot on the third row, second column that as the value of α_0 increases, the height of the acceptance region with respect to the variable γ also increases. Thus, the ratio of the added region which is part of the acceptance region is bigger than that of the original region, which results in an increased acceptance rate. Furthermore, it is interesting to note the similarity of the shapes of Figure 3.10 and the scatterplot of α_0 vs γ, which is, of course, not surprising, since the acceptance rate for a uniform prior depends exclusively on the acceptance region.

However, if the current support S^* already contains the entire acceptance region A_ϵ, it is not difficult to see by choosing any S containing S^*, that the acceptance rate will then decrease, as our intuition dictates. This is given in the next proposition.

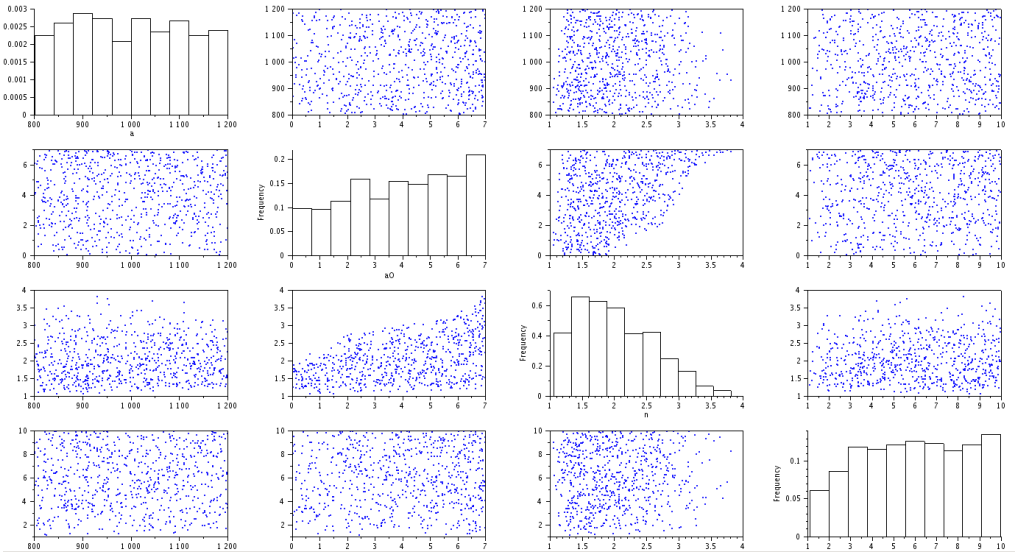


FIGURE 3.11. Scattermatrix of the accepted elements of the sample of our chosen repressilator model. The height of the accepted region in the scatterplot of α_0 vs. γ is increasing as α_0 increases (third row, second column).

PROPOSITION 3.16. *Suppose that the acceptance region A_ϵ is bounded. Then there exists an S^* such that for any $S' \supset S^*$, then $\tau_{S'} \leq \tau_{S^*}$.*

Proof. Choose an S^* which completely contains A_ϵ. Then for any $S' \supset S^*$, $V(S') \geq V(S^*)$, where $V(\cdot)$ represents the volume of the region. Since $V(A_\epsilon)$ remains constant in both cases, it follows from Remark 3.10 that $\tau_{S'} \leq \tau_{S^*}$. □

REMARK 3.17. There are two main reasons why the cardinality of the rejection sample can become 0:

(1) The ϵ is too small, and so the acceptance rate is too small such that the chosen sample size N is not able to obtain enough elements.
(2) The chosen support does not contain any values of θ for which $\rho(\theta) < \epsilon$.

The first case can easily be avoided by choosing an ϵ which is of the appropriate scale with respect to the actual values of the solution for the time points being studied. A method to do this will be introduced in the next section. The second situation will often occur in the case of prior distribution with bounded support, and especially for higher dimensional coefficient estimation. This can be overcome by revising the prior to cover a larger range of values in each variable.

3.5. Improving the method

In this section, we describe three ways to improve the basic rejection method which we have introduced in the previous section.

3.5.1. Sequential Rejection Method. A simple way to increase the acceptance rate would be to choose an improved prior distribution based on the result of a preliminary sample. A natural way to do this would be to begin with a uniform prior for, say, the first 10% of the desired sample. Provided this gives a sufficiently large number of accepted values so that it can adequately represent the acceptance region A_ϵ, we can compute the mean vector μ and the covariance matrix Σ for this sample. Then, for the remaining 90% of the sample, one can change to a multivariate Gaussian distribution with mean and covariance matrix equal to that of the preliminary sample. It turns out that the remaining part has approximately the same acceptance rate, regardless of the value of ϵ, as we shall prove shortly. For this, we first need the following technical result.

PROPOSITION 3.18. *Let Y be a random vector of dimension m whose components are the coordinates of a point chosen uniformly within and on the ellipsoid $y^TAy < 2\epsilon$, where A is positive definite and ϵ is constant. Then the covariance matrix Σ_Y of Y is equal to $\frac{2\epsilon}{m+2}A^{-1}$.*

Proof. Instead of working on the actual ellipsoid directly, assume first that the support is on the unit sphere centered at the origin. More precisely, let $X = (X_1, X_2, ..., X_m)$ be a random vector whose components are the coordinates of a point chosen uniformly within the m-dimensional unit ball $x^Tx < 1$. As the support is symmetric about the origin, the mean of each coordinate and the covariance between any two coordinates must be 0. By Proposition 1.30 and the exchangeability of the X_i's, the variance of each X_i must be $\frac{1}{m+2}$. Thus, the covariance matrix of X is given by $\Sigma_X = \frac{1}{m+2}I_m$, where I_m is the $m \times m$ identity matrix.

Now note that our required ellipsoid $y^TAy < 2\epsilon$ is just a linear transformation of the unit ball. In particular, we have $Y = \sqrt{2\epsilon}A^{-1/2}X$, or equivalently, $X = \frac{1}{\sqrt{2\epsilon}}A^{1/2}Y$. Here, we can compute the inverse and the square root of the matrix since A is positive definite. Thus, the covariance matrix of Y is

$$\begin{aligned}\Sigma_Y &= \sqrt{2\epsilon}(A^{-1/2})^T\Sigma_X\sqrt{2\epsilon}A^{-1/2} \\ &= 2\epsilon(A^{-1/2})^T\frac{1}{m+2}I_mA^{-1/2} \\ &= \frac{2\epsilon}{m+2}A^{-1},\end{aligned}$$

as required. □

An interesting corollary of the previous technical result is that, under some conditions, the acceptance rate for the second part of the sample as we described above, is actually constant even if ϵ is decreased, as we shall now show.

COROLLARY 3.19. *Suppose that the acceptance region A_ϵ is exactly an ellipsoid, and that the uniform distribution on A_ϵ has covariance matrix Σ. If Y is*

drawn from another prior distribution with covariance matrix Σ*, then* $\mathbb{P}(Y \in A_\epsilon)$ *is independent of* ϵ*.*

Proof. By Remark 3.5, μ and Σ are the mean vector and covariance matrix of a uniform distribution within the ellipsoid $(\Delta\theta)^T H(\rho(\theta))\Delta\theta < 2\epsilon$ defined in (3.6). This means that by Proposition 3.18, $\Sigma = \frac{2\epsilon}{m+2}H^{-1}$, or that $H = \frac{2\epsilon}{m+2}\Sigma^{-1}$. Thus, the probability of an arbitrary Y to be within the acceptance region is given by

$$\begin{aligned}\mathbb{P}((\Delta\theta)^T H\Delta\theta < 2\epsilon) &= \mathbb{P}\left((\Delta\theta)^T \frac{2\epsilon}{m+2}\Sigma^{-1}\Delta\theta < 2\epsilon\right) \\ &= \mathbb{P}((\Delta\theta)'\Sigma^{-1}\Delta\theta < m+2),\end{aligned}$$

which is independent of ϵ. □

If we assume that the sample covariance is not too far from Σ and that A_ϵ is close to an ellipsoid, then Corollary 3.19 implies that the strategy of sampling with any distribution with covariance matrix Σ will give the same acceptance rate regardless of the value of ϵ. Thus, if the new distribution is chosen well, this can allow us to obtain a high acceptance rate even if the value of ϵ is small. One such good choice for the distribution is the multivariate Gaussian distribution.

Remark 3.20. If the second part of the sample is drawn from a Gaussian distribution, one can even compute the fixed acceptance rate. Let X be a m-dimensional multivariate Gaussian random vector with mean vector μ and positive-definite covariance matrix Σ. It is known in multivariate analysis that $(X-\mu)'\Sigma^{-1}(X-\mu)$ has the chi-squared distribution with m degrees of freedom. (For a proof, one can refer, for example, to Result 4.7 in Johnson and Wichern [**17**].) Taking the cumulative distribution then allows us to compute $\mathbb{P}(x'\Sigma^{-1}x < k)$ for any constant k.

Example 3.21. We now illustrate the performance of a sequential rejection method in a toy example. As before, consider the repressilator model (see (1.23)) with true coefficients $(\alpha_0, \gamma, \beta, \alpha) = (1, 2, 5, 1000)$, starting point $(0, 2, 0, 1, 0, 3)$, $T = \{0, 0.5, 1, ..., 3\}$, and sample size $N = 5000$. One can observe from the result in Table 3.5 that the average among the five runs of the minimum ρ decreases as ϵ decreases. This is not completely surprising, as we are obtaining approximately the same number of points in a region which contains less points of high distance ρ. Thus, we would intuitively expect the minimum distance to also decrease. Furthermore, we also see that as ϵ decreases in the problem, the acceptance rate remains approximately the same in the sequential method, unlike that of the plain rejection method (for instance, see Table 3.3).

While the method provides a significant improvement from the acceptance rate of the basic rejection sampling algorithm, it also suffers from some problems.

TABLE 3.5. Results using sequential rejection, sample size $N = 5000$, where the first 500 are used as a preliminary sample. The prior distribution is uniform over $[0,7]\times[0,4]\times[0,10]\times[800,1200]$, and the average acceptance results of 5 runs are shown below.

ϵ	Min. distance	Acceptance rate
1200	8.97	66.04
1000	7.67	65.23
800	5.26	62.67
600	4.6	62.45

In practice, we do not know what the shape of the acceptance region A_ϵ is. If A_ϵ is too far from an ellipsoid, the Gaussian prior in the second part of the sample may not be very effective in increasing the acceptance rate. Also, this method is dependent on having a good estimate of Σ. However, if the size of the initial sample is not large enough, the covariance matrix S of this sample may not approximate Σ very well. Furthermore, increasing the size of the sample is not a guarantee of an improved covariance matrix estimate as S is known as an inconsistent estimator for Σ. Despite these problems, performing a sequential method still remains a good choice. We will be looking at a more sophisticated sequential algorithm again later in Section 4.2.

3.5.2. Choosing the value of ϵ. Another important consideration when using the method is choosing an appropriate value of the maximum threshold ϵ. If ϵ is chosen too large, then the method will simply accept all the elements in the sample, making the process useless. On the other hand, if ϵ is too small, then the acceptance rate will be too small, and we run the risk of not getting any accepted elements in the sample. We shall now provide a logical way to choose ϵ based on the support of our prior distribution and the differential system which we shall use to model the data.

To obtain a reasonable value of ϵ, one needs to have an idea of the order of magnitude of $\rho(\theta)$. Suppose that, as before, we take as measure of distance the sum of squared differences

$$\rho(\theta) = \sum_{j=0}^{k} (y(t_j;\theta) - \bar{y}_j)^2,$$

where $\bar{y}$ is assumed to be the model data for $\theta = \theta_0$. Our objective is to construct an estimate of a "typical value" of $\rho(\theta)$.

The main idea of our approximation for ϵ involves computing an estimate of $y(t_j;\theta) - \bar{y}_j = y(t_j;\theta) - y(t_j;\theta_0)$ for a fixed time t_j. Since θ is m-dimensional, a

simple way to do this is to use the total differential

$$y(t_j;\theta) - y(t_j;\theta_0) \approx \sum_{i=1}^{m} \left\{ \frac{\partial y}{\partial \theta^i}(\theta_0) \right\} \Delta\theta^i \tag{3.18}$$

where the superscript i denotes the ith component of the coefficient vector and $\Delta\theta^i = (\theta^i - \theta_0^i)$. The partial derivative in (3.18) can be estimated using a finite difference estimate

$$\epsilon_{ij} = \frac{y(t_j;\theta_0^1,\theta_0^2,...,\theta_0^i+h,...,\theta_0^m) - y(t_j;\theta_0^1,...,\theta_0^m)}{h}. \tag{3.19}$$

where h is a small increment (for example, around 10^{-6}).

We are then left with the choice of θ_0 and $\Delta\theta^i = \theta^i - \theta_0^i$ to be used in the computation. In general, we have no idea of what θ_0 is exactly (otherwise, we would not be estimating it using our method!). The most logical guess we could make is that it is at the center of the support of the prior distribution. Thus, assuming that our prior distribution is a box $(\theta_1^{min},\theta_1^{max}) \times (\theta_2^{min},\theta_2^{max}) \times ... \times (\theta_m^{min},\theta_m^{max})$ on $\mathbb{R}^m$, this gives us the estimate

$$\theta_0^* = \left(\frac{\theta_1^{min}+\theta_1^{max}}{2}, ..., \frac{\theta_m^{min}+\theta_m^{max}}{2} \right).$$

On the other hand, we can think of the size of $\Delta\theta^i$ as an estimate of the maximum difference we are willing to accept between the ith component of θ and the true θ_0. This can be chosen as a fraction p of the length of the interval for that coefficient. That is,

$$\Delta\theta^{i,*} = p(\theta_{max}^i - \theta_{min}^i).$$

Based on our experiments, we recommend choosing p to be between 0.1 and 0.2.

Using the above choices, we are able to obtain an estimate for $y(t_j;\theta) - \bar{y}_j$. Substituting this estimate in $\rho(\theta)$, we end up with the following estimate for ϵ:

$$\epsilon_{est} = \sum_{j=0}^{k} \sum_{i=0}^{m} (\epsilon_{ij}\Delta\theta^{i,*})^2 \tag{3.20}$$

The following example illustrates how ϵ_{est} can be computed in the case of the repressilator (1.23).

Example 3.22. Consider the repressilator model (1.23), where we assume that the initial point is at $(0,2,0,1,0,3)$ and the model coefficients to be $(\alpha_0,\gamma,\beta,\alpha) = (1,2,5,1000)$. Suppose the known data is at the times $T = \{0,1,2,...,7\}$, and that the prior distribution is uniformly distributed on $[0,2]\times[1.5,2.5]\times[0,10]\times[900,1100]$. Also, we choose $p = 0.1$ Then $\theta_0^* = (1,2,5,1000)$ and the value of $\Delta\theta^{i,*}$ for $i = 1,2,3,4$ are 0.2,0.1,1, and 20, respectively. A series of finite difference computations would then allow us to compute $\epsilon_{est} \approx 865$. This gives approximately a 1% acceptance rate.

We shall revisit this method of choosing ϵ in a more complicated example later in Section 3.6.3.

3.5.3. Another best estimate. Even if the choice of the minimum distance estimate introduced in Section 3.4.2 is usually better than the average, we lose in taking the minimum distance estimate the benefit of taking an average between several good estimates in order to average the different errors and get something in the interior of all t he estimates. This observation gives the idea of a new way to deduce an estimate of the coefficient from the rejection sample, that we will now explain and illustrate with an example.

The main idea is the following: if we average the d best values of the coefficients θ in terms of the distance ρ, and if we increase the number of values in this average, the result will initially be located around θ_0 before it tends, as d increases, towards the mean of the distribution $\pi(\theta|\rho(\theta) < \epsilon)$. Thus, taking the mean of these few initial θ's can often give a better result than just choosing the one with the minimum distance.

More precisely, let $\theta_1, \theta_2, ..., \theta_n$ be a rejection sample. Denote by $\rho(\theta_{(1)})$,$\rho(\theta_{(2)})$, ...,$\rho(\theta_{(n)})$ the distances $\rho(\theta_1), \rho(\theta_2), ..., \rho(\theta_n)$ arranged in increasing order. This consequently defines the reordering of the sample from the best (lowest ρ) to the least $\theta_{(1)}, \theta_{(2)}, ..., \theta_{(n)}$. Let

$$\bar{\theta}_d = \frac{1}{d}\sum_{i=1}^{d} \theta_{(i)}$$

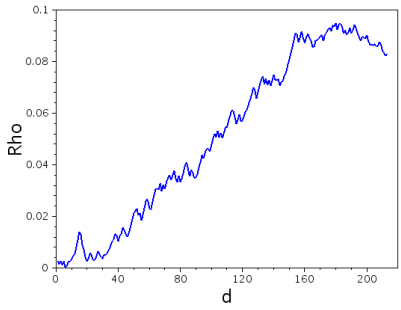
for some d, $1 \le d \le n$. The *least mean estimator* is given by

$$\widehat{\theta}_{leastmean}(d) = \underset{\theta \epsilon \{\bar{\theta}_1, ..., \bar{\theta}_d\}}{\text{ArgMin}}\ \rho(\theta). \tag{3.21}$$

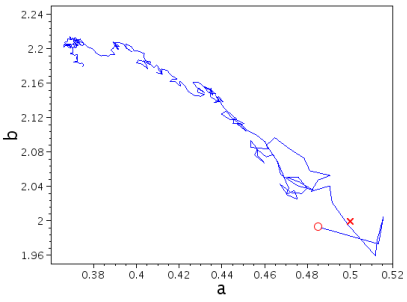
After computing $\widehat{\theta}_{leastmean}(d)$ for a large number of numerical experiments, it has been observed that in most cases, this estimator produces coefficients with the least distance when $d = 10$. Also, we note that the minimum distance estimate $\hat{\theta}_{min}$ is equal to $\widehat{\theta}_{leastmean}(1)$.

EXAMPLE 3.23. We apply this choice of best estimate to compute a best guess of θ in the simplified Lotka-Volterra model introduced in (1.22). Figure 3.12 shows the evolution of $\widehat{\theta}_{leastmean}$ in estimating the coefficients of a Lotka-Volterra model as d increases.

In Figure 3.12a, we observe that when d increases, $\rho(\widehat{\theta}_{leastmean}(d))$ initially decreases, and then increases until it stabilizes at a certain point. By the Law of Large Numbers, this point is the value of ρ for the center of A_ϵ. Figure 3.12b illustrates why $\rho(\widehat{\theta}_{leastmean}(d))$ decreases initially before it increases again. The coefficients $\bar{\theta}_d$ are initially located around θ_0 for small values of d, before they tend towards the center of A_ϵ as d increases.



(A) Typical graph of $\rho(\widehat{\theta}_{leastmean}(d))$ as d increases



(B) Evolution of $\widehat{\theta}_{leastmean}(d)$ as d increases. The red circle represents the minimum distance point, and the red 'x' are the model coefficients.

FIGURE 3.12. Evolution of $\widehat{\theta}_{leastmean}(d)$ as the number of elements d increases for the coefficients in a Lotka-Volterra model with model coefficients $(a, b) = (0.5, 2)$. The initial point of the model data is at $(1, 0.5)$ and the time points are at $T = \{0, 0.5, 1, ..., 3\}$. The prior distribution is uniform over $[0, 3] \times [0, 3]$.

By choosing $\widehat{\theta}_{leastmean}(10)$, we get a substantial improvement in the distance of the estimate. In this particular run that we did, the least distance is when $\theta = \bar{\theta}_6$, where $\rho = 0.0001$, vs. $\rho = 0.0028025$. While the result will not always be the same, we will obtain an improvement by choosing $\widehat{\theta}_{leastmean}(10)$ than $\widehat{\theta}_{leastmean}(1)$ most of the time.

Furthermore, empirical results suggest that this strategy becomes even more effective when estimating the coefficients in a differential system with a higher dimension of the coefficient space. In our repressilator model with 6 unknown coefficients, in 25 out of 30 runs, the absolute maximum for $\rho(\bar{\theta}_{leastmean})$ occurs within the first 10 means. In all of the runs, the "minimum distance" estimator is improved. In Comet et.al.'s simplified circadian cycle system (with 12 unknown variables) which we introduced in Section 1.4.5, in all 12 runs made, the absolute minimum occurs within the first 5. In 9 out of 12 of these runs, the minimum distance estimator is improved.

3.5.4. Implementing the Method. Although the method which we have described is quite simple, one will quickly realize that there are a large number of parameters that need to be chosen when implementing it. Without a careful choice of these parameters, one can easily obtain a rejection sample of size 0. As such, we now provide a short summary of the steps needed to apply the rejection

method to produce a distribution of coefficients of a differential system that best fits some known data.

(1) Choose an initial prior distribution π_0 for the unknown coefficients, and an initial sample size.
(2) Choose a proper ϵ. To do this, one can use the procedure outlined in Section 3.5.2, and choose the percentage of the range to be around 10 to 20 percent. The resulting acceptance rate will vary depending on the differential system, but empirical results suggest it will be at least 3%.
(3) Provided we obtain a reasonably large rejection sample in the previous step, change the prior distribution to a Gaussian distribution, with a mean equal to the sample mean and covariance matrix equal to a fraction of the sample covariance (around 10% would be a typical choice).
(4) After we obtain the rejection sample $\theta_1, \theta_2, ..., \theta_n$, the scattermatrix of these samples can be obtained to visualize the estimated law of θ. However, if a point estimate is required, arrange the θ_i's in increasing order based on the metric ρ. Then compute $\widehat{\theta}_{leastmean}(10)$ based on the procedure introduced in Section 3.5.3.

3.6. Application to perturbed model data

In this section, we examine the robustness of the rejection method when estimating the coefficients of a differential system using perturbed "model" data. This means that our known data $\bar{y}$ no longer corresponds exactly to the discrete trajectory for a particular value of the coefficients θ. We shall call such perturbed data more simply as *noisy data*. Dealing with noisy data is more realistic as there is no such thing as "exact data." This is not only because measurement errors are unavoidable in real-life data, but mainly because any differential system used to model a natural phenomenon will necessarily be an oversimplification of reality, as it is essentially impossible to capture all the factors at work in a certain phenomenon. Even when it is possible to do so, this would entail an unnecessarily complicated system having a very large numbers of equations and coefficients. Thus, while a simpler system would not be able to capture reality exactly, it will still be useful in understanding the dynamics of the process.

This section is comprised of three parts. In the first part, we give some comments on how to generate noisy data and why we decided not to run most of our experiments using noisy data. The remaining part of the chapter can be divided into two parts. In section 3.6.2, we shall go back to each of the properties of the rejection method introduced in section 3.4 and verify whether each of these continues to hold for noisy data. Then, we shall apply our method to fit the coefficients of the circadian cycle model introduced in Section 1.4.5 based on simulated noisy data.

3.6.1. Some comments on noisy data. For all our discussion in this chapter so far, we have decided to use "exact data", that is, where the given data corresponds to the trajectory of the differential system for a fixed value of $\theta = \theta_0$, instead of using simulated noisy data. This choice was intentional, and we shall explain why very shortly. To do this, we first briefly explain how we can produce theoretical noisy data.

There are many ways to produce theoretical perturbed data. Here, we shall only mention two possibilities. As before, let $y' = g(y;\theta)$ be the differential system used to model the data, where θ consists of the unknown coefficients to be estimated. Assume that the known data are at times $T = \{t_0, t_1, ..., t_k\}$, with values $\bar{y}(T) = (\bar{y}_0, ..., \bar{y}_k)$.

(1) A first option is to incorporate an additive error to each component of $\bar{y}(T)$. Usually, this additive error is represented as a Gaussian vector $\epsilon = (\epsilon_1, \epsilon_2, ..., \epsilon_k)$, where each ϵ_i is multivariate Gaussian with mean 0 and a fixed covariance matrix. Thus, the original model data $\bar{y}(T)$ becomes $\bar{y}(T) + \epsilon$. This can be interpreted as the typical measurement error for our known data.
(2) A second possibility is to perturb the coefficient θ. This can be done in several ways. One option is to make the model data $\bar{y}(T)$ be the solution of $y' = g(y;\theta + \epsilon)$ for an error term ϵ with the appropriate dimensions. Another, more general, way could be to choose a distribution of θ around the "true" value θ_0. Take a sample of size k for θ from this distribution. Then, to compute $\bar{y}(t_k)$, we compute the value of the solution at the time t_k for the differential equation $y' = g(y;\theta)$, but with boundary condition $y(t_{k-1}) = \bar{y}(t_{k-1};\theta_{k-1})$, where $\bar{y}(t_{k-1};\theta_{k-1})$ is the point obtained using $\theta = \theta_{k-1}$. This type of error can be encountered for example when there are varying environmental conditions as time elapses, like when the temperature of the environment is changing.

For example, consider the competing species model,

$$\frac{dy_1}{dt} = ay_1 - y_1^2 - 0.5y_1y_2 = y_1(a - y_1 - 0.5y_2)$$

$$\frac{dy_2}{dt} = by_2 - 0.5y_2^2 - 1.5y_1y_2 = y_2(b - 0.5y_2 - 1.5y_1)$$

where $T = \{0, 0.5, 1, ...3\}$, initial point $(x, y) = (1, 0.5)$, and model coefficients $(a, b) = (1, 1.5)$. The corresponding graph of the phase plane of the differential equation is given in Figure 3.13.

Assuming a Gaussian perturbation with mean 0 and standard deviation 0.1, the typical graphs of the perturbed data using the methods mentioned above are given in Figure 3.14. One can see that the dynamics of the trajectory may

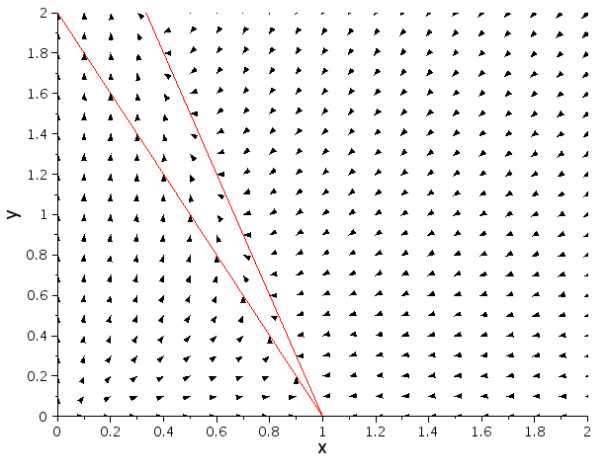
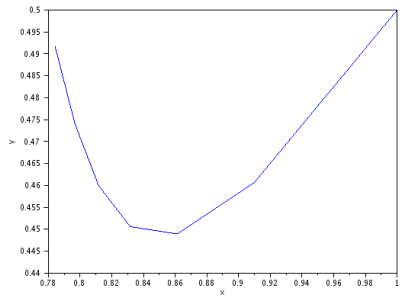


FIGURE 3.13. Phase plane of the competing species model where $a = 1$ and $b = 1.5$. The red lines correspond to the graphs of $a - y_1 - 0.5y_2 = 0$ and $b - 0.5y_2 - 1.5y_1 = 0$. The separatrix is somewhere in the region between the two lines.

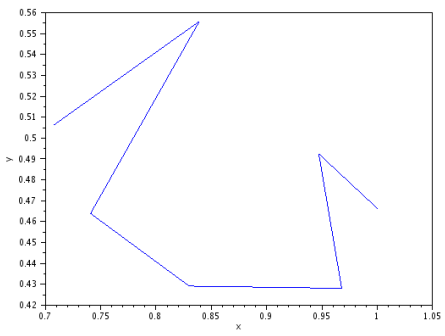
become completely different. This is because perturbing the data or the coefficients may either cause the initial point to cross the separatrix, the target θ to move, or the location of the separatrix to move. Any of these cases may alter the trajectory drastically.

As with any coefficient estimation method, it is clear that if the amount of simulated noise is too large, it will be difficult, or even impossible, to recover the unperturbed value of the coefficient θ using any coefficient estimation method with a reasonable degree of accuracy. This is partly because the shape of the resulting discrete trajectory may become drastically different from the possible trajectories of the proposed differential model. However, this is not a cause for concern in general, as this becomes a problem with the choice of model, instead of the coefficient estimation method. As we will discuss further in the next section, if the perturbation is small enough so as to maintain the general shape of ρ, the resulting estimate for θ will not be drastically different. Hence, studying the method using exact data will not give us significantly different results than with these noisy data. An additional advantage with using exact data is that it allows us to check whether inaccuracies in the estimate are due to the method itself rather than the size of the perturbation.

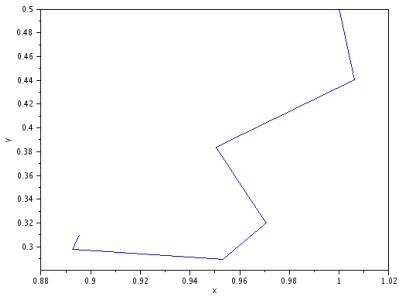
3.6.2. Re-examining the properties of the rejection method. In section 3.4, we examined some properties of the rejection method when we were trying to estimate the coefficients of a differential system based on known data produced from a specific value θ_0 of the coefficients. We now re-examine the same properties, but this time, for noisy data.



(A) Actual graph of the competing species model for $a = 1, b = 1.5$, and initial point $(1, 0.5)$.



(B) Effect of perturbing the data using option (1)



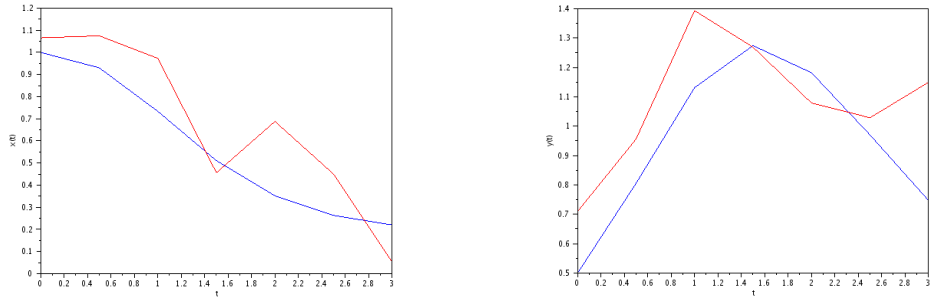
(C) Effect of perturbing the data using option (2)

FIGURE 3.14. One possible scenario after the two types of perturbation are applied to the competing species model.

In Remark 3.5, we saw that the acceptance region becomes approximately an ellipsoid if ϵ is small enough. Recall that the result was obtained by writing out the Taylor expansion of $\rho(\theta)$ for the differential equation $y' = g(y;\theta)$ given in (3.2). When the known data does not correspond exactly to a specific value of θ, this will continue to be true as long as g (and therefore ρ) is of class C^2, and the minimum value of ρ is achieved for a value of θ in the interior of S. In this case, there will still be a value θ_0^* which gives the minimum ρ and one can still compute the Taylor expansion of ρ, as given before. However, note that the minimum ρ will not usually be equal be 0. Thus, it is possible to have no accepted values if the chosen ϵ is too small. Also, this value θ_0^* will no longer be the value of θ corresponding to the unperturbed case.

Next, we note that Proposition 3.8 still holds true, and the minimum distance estimator $\hat{\rho}_n$ defined before still converges in probability to the minimum distance $\rho(\theta_0)$. However, since the minimum ρ is no longer 0, it is no longer clear that the θ_0 with minimal distance gives us the best estimate for θ. We give one such example on a very simple Lotka-Volterra coefficient estimation problem.

EXAMPLE 3.24. *Suppose we wish to estimate the coefficients in the two-coefficient Lotka-Volterra model described in Section 1.4.3. We assume that the true coefficients are* $\theta = (a, b) = (0.5, 2)$*, the initial point is* $(1, 0.5)$*, and the time points* $T = \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$*. The prior distribution is assumed to be uniform over* $[0, 3] \times [0, 3]$*. A set of perturbed data using option (1) is shown in Figure 3.15.*



(A) (B)

FIGURE 3.15. Graphs of $x(t)$ (A) and $y(t)$ (B) for the original data (in blue) and noisy data (in red). The noisy data were produced by adding a Gaussian term with mean 0 and standard deviation 0.2 to each coordinate in the original trajectory.

In this case, the resulting contour map for $\rho(\theta)$ *is given in Figure 3.16. One can see that for the small values of* ϵ *provided, the shape of the region enclosed by* A_ϵ *remain approximately ellipses. However, the coefficients* θ *which give the minimum* ρ *have shifted to around* $(0.6, 1.9)$*. If we run the rejection method for* $\epsilon = 1$*, notice that the center of the ellipse is closer to our target value* $(0.5, 2)$ *than the coefficients that give the minimum* ρ*, so the mean appears to be the better estimate in this case.*

Unfortunately, the values $y(t_i)$, $i = 0, 1, 2, ..., k$ are not independent. However, if they were, we can gain some insight as to what will be the new minimum point θ_0^*. To do this, one may look at the distribution of $\rho(\theta)$. Assuming that the error for each data point is Gaussian with mean 0 and variance σ^2, each term $y(t_i; \theta) - \bar{y}$ is now Gaussian with mean $y(t_i; \theta) - \bar{y}$ and variance σ^2. If each of these k differences can be assumed to be independent, then $\rho(\theta)/\sigma^2$ has a noncentral chi-squared distribution with k degrees of freedom, with noncentrality coefficient

$$\lambda = \sum_{i=1}^{k} \left(\frac{y(t_i; \theta) - \bar{y}}{\sigma} \right)^2.$$

Finally, since all the acceptance rate results we obtained are independent of the data $\bar{y}$, these still apply for the case of inexact data. However, for a fixed

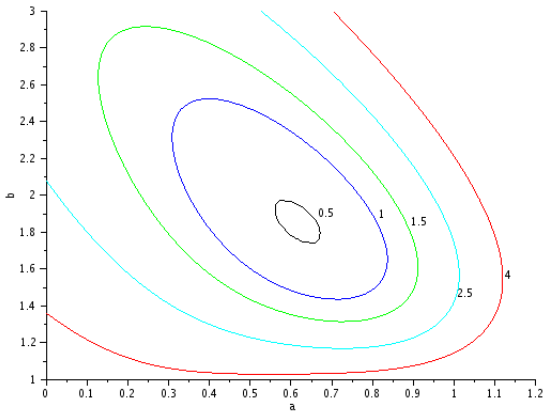
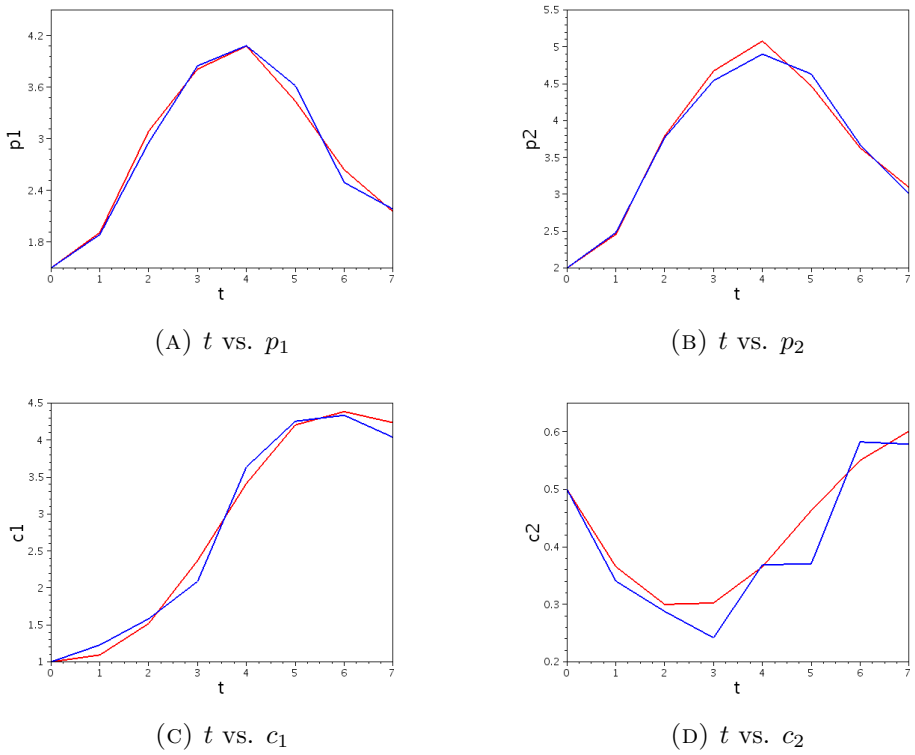


FIGURE 3.16. Contour map of $\rho(\theta)$ when estimating θ in a Lotka-Volterra model assuming the known data in Figure 3.15

value of ϵ, the acceptance region A_ϵ will be of smaller size than before. Thus, the corresponding acceptance rate will also be smaller. This further aggravates the acceptance rate problem for systems with a large number of dimensions, as we recall from Proposition 3.12 that the decrease in acceptance rate is of order $l/2$ as the dimension l increases.

3.6.3. Application of the method. We shall now use the rejection sampling method and the steps outlined in Section 3.5.4 to estimate the coefficients in simplified circadian cycle model which we introduced in Section 1.4.5. Recall that the system consists of 4 differential equations and 12 variables.

We begin by generating the model data for the differential system (1.24). The coefficients chosen were $K = 0.4, \gamma = 15, k_1 = 0.08, k_2 = 0.06, k_3 = 0.08, k_4 = 0.06, kd_1 = 0.05, kd_2 = 0.05, kd_3 = 0.05, kd_4 = 0.45, v_1 = 2, v_2 = 2.2$. Here, we obtain oscillations which have decreasing amplitudes, as the value of kd_4 is greater than 0.412182, the bifurcation point as computed in [**3**]. However, the results remain largely unchanged even if we choose $kd_4 < 0.412182$. Graphing the solution of the differential equation for the chosen coefficients, one can show that the first complete oscillation occurs after around every 12 units. Since there is no reason to believe that the data which we will obtain will correspond exactly to one "period", we assume that we are given the discrete trajectory at time points $0, 1, 2, ..., 7$, which corresponds to around half a period. We incorporate a small measurement error by adding a Gaussian term with mean 0 and standard deviation 0.15 for the first three variables, and mean 0 and standard deviation 0.05 for the last variable. The standard deviations were chosen to be different for each variable to take into account the approximate sizes of the values of



(A) t vs. p_1 (B) t vs. p_2

(C) t vs. c_1 (D) t vs. c_2

FIGURE 3.17. Graphs of the "true" trajectories (in red) and the perturbed trajectories (in blue) for each of the 4 variables.

each variable. A comparison of the model data and perturbed data is given in Figure 3.17.

The prior distribution is chosen to be uniform, where the lower and upper bounds are given in the table below:

Variable	Minimum in prior	Maximum in prior	Reference value
K	0	1	0.4
γ	0	20	15
k_1	0	0.2	0.08
k_2	0	0.2	0.06
k_3	0	0.2	0.08
k_4	0	0.2	0.06
kd_1	0	0.2	0.05
kd_2	0	0.2	0.05
kd_3	0	0.2	0.05
kd_4	0	0.5	0.45
v_1	0	5	2
v_2	0	5	2.2

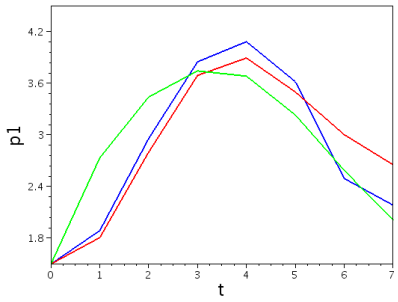
We shall use the guidelines we proposed in Section 3.5.4 in running the rejection and sequential rejection method. First, we shall choose $\epsilon \approx 27$, which corresponds to the value obtained when we use the total differential estimate in Section 3.5.2 with θ as the center of the support, $h = 10^{-6}$, and $p = 0.12$. We shall do two sets of ten runs, the first set being a basic rejection method with 10000 iterations. For the second set of ten runs, we shall use the sequential rejection method introduced in the previous section, with an initial sample of size 2000. We then run another 10000 iterations, but now with a multivariate Gaussian distribution with mean and covariance equal to the corresponding mean and covariance of the initial sample. We shall take as point estimate $\widehat{\theta}_{leastmean}(10)$, where we chose the minimum among the first 10 averages.

The results using both the basic rejection method and the sequential rejection method are given in Tables 3.6. Aside from the improved acceptance rate, we can see a substantial improvement in the average distance of our point estimate to the given perturbed data.

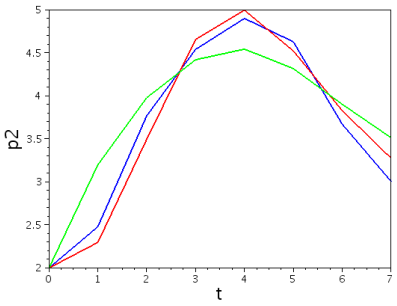
TABLE 3.6. Results when applying the rejection method, with and without the sequential improvement, to estimate the coefficients in Comet's simplified circadian cycle model. For each of the ten runs, $\widehat{\theta}_{leastmean}(10)$ and the corresponding distance ρ is computed.

	Sequential rejection for $\epsilon = 27$	Basic rejection for $\epsilon = 27$
Average acceptance	50.46%	3.487%
Average distance	2.13	3.25

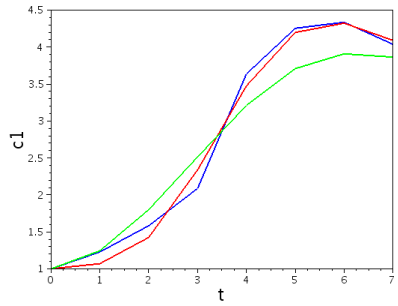
An even more interesting comparison is provided by Figure 3.18, where we graph the trajectories resulting from the "best guess" coefficient estimates using the sequential rejection method with the actual data. We graph the trajectories resulting from two different values of θ obtained from the 10 runs: the value of $\widehat{\theta}_{leastmean}(10)$ that produced the smallest ρ (in red), and the one which produced the biggest ρ (in green). These represent the best and worst scenarios when estimating the coefficients using the sequential rejection method. We can see that even with the limited amount of information provided by eight points of data, the method can give coefficients that fit the known data well, even in a twelve dimensional problem.



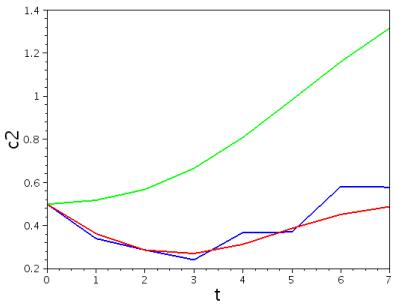
(A) Graph of t vs. p_1



(B) Graph of t vs. p_2



(C) Graph of t vs. c_1



(D) Graph of t vs. c_2

FIGURE 3.18. Graphs of the resulting trajectories using the worst and best coefficients (as given in the previous table) using the sequential rejection method. The graphs in red are using the "best" coefficients, the ones in green are the "worst" coefficients, and the ones in blue are those of the perturbed data.

CHAPTER 4

Estimating coefficients of systems of differential equations: further approaches

In this chapter, we discuss two alternative ways to sample from and explore the acceptance region A_ϵ. Both methods discussed use "local moves" where the next sample is chosen in a neighborhood of the previous accepted value, unlike that of the rejection sample (or RS for short). While these methods still target the same region A_ϵ, we shall see how these can produce a larger acceptance rate and avoid many of the disadvantages of the RS method discussed in the previous chapter. This will also allow us to be able to handle higher dimensional coefficient spaces.

4.1. A Markov chain Monte Carlo method

An interesting old algorithm, Markov Chain Monte Carlo (or MCMC for short), introduced by Metropolis et. al. in 1953, will allow us to improve the RS algorithm by increasing effectively the acceptance rate. In this section, we will provide a detailed exposition on an MCMC-based algorithm to estimate the coefficients in a system of differential equations. After a short presentation of the method, we shall prove its theoretical validity. Several properties of the method are then provided, which are illustrated by simulation examples.

4.1.1. Presentation of the Algorithm. The method which we shall introduce is based on an MCMC scheme introduced by Marjoram et. al. in [**21**]. In the MCMC technique, we wish to sample from a possibly intractable distribution $\pi(\theta)$. To do this, we perform a random walk inside the support of π by generating a trajectory of a Markov Chain which has $\pi(\theta)$ as stationary distribution. Theorem 1.25 then guarantees that provided the Markov chain is irreducible and aperiodic, the law of the resulting trajectory converges in the total variation norm (recall equation (1.9)) to $\pi(\theta)$. In addition, we are also assured that the average of the first n values in the trajectory converges almost surely to the expected value of θ as $n \to \infty$ (see Theorem 17.0.1 in [**27**]).

To estimate the coefficients of a system of differential equations, we apply the same concept, but now to find the same posterior distribution $\pi_\epsilon(\theta|\bar{y})$ as in the rejection method in the previous chapter. Let $y' = g(y;\theta)$, $\bar{y}(T)$, $\pi_0(\theta)$, S_0, and $\rho(\theta)$ be defined as in the previous chapter. We begin with an initial

guess θ_0 of the coefficients for $i = 1, 2, \ldots$. Instead of drawing coefficients θ_{i+1} independently of θ_i from π_0, we now choose a *proposal distribution* $q(\theta_i, \theta^*)$, which defines the conditional distribution of the next proposed sample θ^* given the current sample θ_i. In any case, for reasons which will be made clear shortly, the proposal distribution must be chosen such that $q(\theta^*, \theta_i) > 0$ if and only if $q(\theta_i, \theta^*) > 0$, which is satisfied in the case of a Gaussian q. For example, one can choose $q(\theta_i, \theta')$ to be the kernel of a Gaussian distribution with mean θ_i and a fixed covariance matrix Σ.

We produce our sequence of coefficients $(\theta_i)_{i \in \mathbb{N}}$ recursively as follows. If the proposed sample θ^* satisfies $\rho(\theta^*) < \epsilon$, then the probability to accept θ^*, that is, the probability for θ_{i+1} to be set to θ^*, is defined by

$$\alpha(\theta_i, \theta^*) = \frac{\pi_0(\theta^*) q(\theta^*, \theta_i)}{\pi_0(\theta_i) q(\theta_i, \theta^*)} \mathbb{1}_{\{\rho(\theta^*) < \epsilon\}} \wedge 1. \tag{4.1}$$

where the $\wedge$ denotes the minimum between the two quantities. Otherwise, θ_{i+1} remains equal to the previous value θ_i. Note that α is well-defined because of the small condition that we imposed on the proposal distribution q.

Two special cases of α need to be emphasized. First, if the proposal distribution q is symmetric around its mean, then the acceptance probability α reduces to

$$\alpha(\theta_i, \theta^*) = \frac{\pi_0(\theta^*)}{\pi_0(\theta_i)} \mathbb{1}_{\{\rho(\theta^*) < \epsilon\}} \wedge 1$$

In this case, one has an easy interpretation of the acceptance criterion – assuming $\rho(\theta^*) < \epsilon$, one always accepts the proposed sample θ' if it has a greater likelihood than θ_i based on π_0. Otherwise, one still has a probability of accepting θ^*, but proportional to the ratio of the likelihoods of θ^* and θ_i. Secondly, if in addition to q being symmetric around its mean, the prior distribution π_0 is also uniformly distributed on its support, then the second condition always holds. In this case, the MCMC algorithm will accept any θ such that $\rho(\theta) < \epsilon$. Thus, aside from having a fixed starting point and the "prior" distribution which varies in each iteration, we see that its acceptance condition becomes the same as the rejection method in the previous chapter. We shall initially look at the method in this simplest case later, and then afterwards, examine the effect of choosing a non-uniform prior distribution on the sample.

To help in understanding how the method works, we now take a look at a very simple example.

EXAMPLE 4.1. We apply the MCMC method to estimate the coefficients r and K in the logistic differential equation $y' = ry(1 - y/K)$, and using the same data as in Example 3.1. For the moment, we consider the simplest case, where the prior distribution π_0 is uniformly distributed over $[0, 1] \times [100, 300]$ and the proposal distribution is chosen as multivariate Gaussian, centered around the

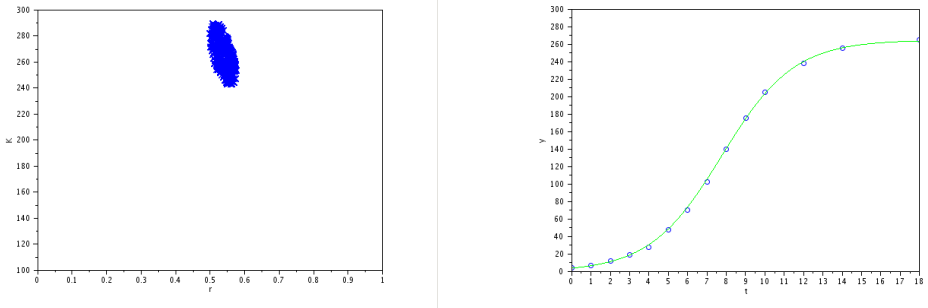
previous value θ_i. That is,

$$\pi(\theta_i, \theta^*) = \frac{1}{2\pi|\Sigma|^{-1/2}} e^{-\frac{1}{2}(\theta^* - \theta_i)'\Sigma^{-1}(\theta^* - \theta_i)},$$

where we assume at the moment that

$$\Sigma = \begin{pmatrix} 0.1 & 0 \\ 0 & 3600 \end{pmatrix},$$

which is equivalent to a standard deviation of slightly under 1/3 the interval of each variable. We assume that the starting point is exactly in the middle of the support; that is, $(r_0, K_0) = (0.5, 200)$. Finally, as in Example 3.4, the maximum threshold is $\epsilon = 1300$. For this particular run, we obtained 307 accepted elements. The mean of the coefficients in the trajectory is $(0.5373, 266.43)$, while the θ_i that gives the minimum distance is $(0.5381, 264.99)$, both of which are very close to the ones obtained in Example 3.4. The graph of the resulting sample is shown in Figure 4.1. Notice that the acceptance region in Figure 4.1a is virtually the same as before, except that we have a higher number of points within A_ϵ. Figure 4.1b shows the curve of the solution to the logistic equation corresponding to the value of θ which gives the smallest ρ (the "minimum distance" coefficients).



(A) (B)

FIGURE 4.1. Results for logistic model using the MCMC algorithm. The first figure (left) is a plot of the resulting sample in the rectangle which is the support S_0 of the prior π_0. The second figure (right) is the graph of the solution of the logistic equation with "minimum distance" coefficients as previously described.

We shall see in Section 4.1.2 that under certain conditions, as the length of the chain increases, the elements produced in one trajectory of this method will have as distribution $\pi_\epsilon(\theta|\bar{y})$, which is the same as that of the rejection method introduced in the previous chapter. Also, we shall introduce a way to choose both the starting guess and the covariance matrix Σ of the proposal distribution more systematically.

4.1.2. Computation of the posterior distribution. In this section, we shall show that we can generate a Markov Chain which has $\pi_\epsilon(\theta|\bar{y})$ as its stationary distribution. This will be done in two steps. First, we shall prove that the sample produced by the algorithm is a trajectory of a Markov chain. Then, we shall show that this Markov chain has a stationary distribution, and that it converges to this distribution. We begin by formalizing the MCMC algorithm that we introduced as a sequence of random variables $(X_n)_{n\in\mathbb{N}}$.

DEFINITION 4.2. *Suppose that q is a transition kernel on $\mathbb{R}^m \times \mathcal{B}(\mathbb{R}^m)$, $\theta_0 \in S_0$, $X_0 = \delta_{\theta_0}$. Let $(U_n)_{n\geq 1}$ be an iid sequence of random variables uniformly distributed on $(0,1)$ independent of X_0, and define the sequence $(X_n)_{n\geq 1}$ of random variables on S_0 as follows:*

$$X_{n+1} = q(X_n,\cdot)\cdot \mathbb{1}_{\{U_{n+1}<\alpha(X_n,X_n^*\}} + X_n \cdot \mathbb{1}_{\{U_{n+1}\geq\alpha(X_n,X_n^*)\}}, \tag{4.2}$$

*where $X_n^* = q(X_n,\cdot)$. Any $X_1, X_2, ..., X_n$ generated this way is called an **MCMC sample**.*

This defines the same random sequence as the MCMC algorithm that we described in the previous section. Indeed, the first term sets $X_{n+1} = X_n^*$ with probability $\alpha(X_n, X_n^*)$, while the second term makes X_{n+1} equal to the previous value X_n as the complement with probability $1-\alpha(X_n, X_n^*)$.

THEOREM 4.3. *The MCMC sequence defined by (4.2) is a trajectory of a Markov chain of kernel*

$$K(x,z) = \alpha(x,z)q(x,z) + \left(1 - \int_{x\neq z} \alpha(x,z)q(x,z)dz\right)\mathbb{1}_{\{x=z\}} \tag{4.3}$$

Proof. To show that the sequence $(X_n)_{n\in\mathbb{N}}$ is a Markov chain, note that X_{n+1} can be expressed in the form $h(X_n, Y_{n+1})$, where $Y_{n+1} = U_{n+1}$. Thus, by Lemma 1.19, $\{X_n\}$ is indeed a Markov chain.

Next, we compute the kernel of this Markov chain. Let A be any subset of S_0. Then

$$\begin{aligned} K(x,A) &= \mathbb{P}(X_{n+1}\in A|X_n = x) \\ &= \mathbb{P}(Y\in A \text{ and } X_{n+1} = Y|X_n = x) + \mathbb{P}(x\in A \text{ and } X_{n+1} = x|X_n = x) \\ &= \int_A q(x,y)\alpha(x,y)dy + \int_{\mathcal{Y}} \mathbb{1}_{\{x\in A\}}(1-\alpha(x,y))q(x,y)dy \end{aligned}$$

where $\mathcal{Y}$ is the support of q. Taking the limiting case when $A = \{z\}$ gives the desired result. □

In the next theorem, we will show that the Markov chain which we have introduced in Theorem 4.3 has $\pi_\epsilon(\theta|\bar{y})$ as stationary distribution.

THEOREM 4.4. *Let $\pi_\epsilon(\theta|\bar{y}) = \frac{\pi_0(\theta)\mathbb{1}_{A_\epsilon}(\theta)}{\int_{A_\epsilon}\pi_0(\theta)d\theta}$, where $A_\epsilon = \{\theta\in S_0|\rho(\theta)<\epsilon\}$, and any kernel of a Markov chain $q(\theta,\theta^*)$ satisfying the condition that $q(\theta,\theta^*)>0$ if*

and only if $q(\theta^*, \theta) > 0$. *Then the distribution* $\pi_\epsilon(\theta|\bar{y})$ *is a stationary distribution of the Markov chain defined by* (4.2).

Proof. By Lemma 1.24 and Theorem 4.3, it suffices to show that

$$\pi_\epsilon(\theta|\bar{y})K(\theta,\theta^*) = \pi_\epsilon(\theta^*|\bar{y})K(\theta^*,\theta) \tag{4.4}$$

If $\theta = \theta^*$, the equality follows trivially. Therefore, assume that $\theta \neq \theta^*$. Assume first that $\alpha(\theta,\theta^*) < 1$. Then $\alpha(\theta^*,\theta) = 1$ and

$$\begin{aligned}
\pi_\epsilon(\theta|\bar{y})K(\theta,\theta^*) &= \pi_\epsilon(\theta|\bar{y}) \cdot q(\theta,\theta^*)\alpha(\theta,\theta^*) \\
&= \pi_\epsilon(\theta|\bar{y}) \cdot q(\theta,\theta^*) \cdot \frac{\pi_0(\theta^*)q(\theta^*,\theta)}{\pi_0(\theta)q(\theta,\theta^*)} \mathbb{1}_{\{\rho(\theta^*)<\epsilon\}} \\
&= \frac{\pi_0(\theta)}{\int_{A_\epsilon} \pi_0(\theta)d\theta} \cdot q(\theta,\theta^*) \cdot \frac{\mathbb{1}_{A_\epsilon}(\theta^*)\pi_0(\theta^*)q(\theta^*,\theta)}{\pi_0(\theta)q(\theta,\theta^*)} \\
&= \frac{\pi_0(\theta^*)\mathbb{1}_{A_\epsilon}(\theta^*)}{\int_{A_\epsilon} \pi_0(\theta^*)d\theta^*} \cdot q(\theta^*,\theta) \\
&= \pi_\epsilon(\theta^*|\bar{y})K(\theta^*,\theta)
\end{aligned}$$

as $K(\theta,\theta^*) = q(\theta,\theta^*)$ when $\theta \neq \theta^*$ and $\alpha(\theta,\theta^*) = 1$. The proof for the case when $\theta \in A_\epsilon$ and $\alpha(\theta,\theta^*) = 1$ follows in a similar manner.

□

While the preceding theorem guarantees that the Markov chain defined by (4.2) has indeed $\pi_\epsilon(\theta|y)$ as stationary distribution, we still need to make sure that the Markov chain converges to the said distribution. If we want to use this Markov chain to produce a sample that will approximate the distribution $\pi_\epsilon(\theta|\bar{y})$, then q must be chosen so that the Markov chain is π_ϵ-irreducible and aperiodic (for example, if q is Gaussian). In this case, Theorem 1.25 guarantees that the law of the MCMC sequence X_n converges in the total variation norm (recall Equation 1.9) to the stationary distribution $\pi_\epsilon(\theta|\bar{y})$.

Since the MCMC method uses local moves instead of randomly choosing around the support of the prior distribution, one can imagine that a larger percentage of the sample would tend to fall into the acceptance region A_ϵ. This is true in general, provided that the perturbation provided by the proposal distribution is small enough. However, to be able to describe exactly this phenomenon, we need to first define the MCMC acceptance rate.

Definition 4.5. *Let* $X_0, X_1, X_2, ..., X_n$ *be an MCMC sample and let* $I_i : \mathbb{1}_{\{X_i \neq X_{i-1}\}}$ *be the corresponding sequence of i.i.d. Bernoulli random variables. The acceptance rate of this sample is given by*

$$\tau_{MCMC} = \frac{1}{N}\sum_{i=1}^{N} I_i.$$

4.1.3. Properties of an MCMC Sample. In this section, we examine the properties of an MCMC sample, and compare it to the rejection sample which we introduced in the previous chapter.

Choosing Σ and the proposal distribution

Central to the success of an MCMC trajectory is the proper choice of the proposal distribution. The most natural choice would be for q to be Gaussian, and centered on the previous value of θ, and with a covariance matrix of Σ. In this case, one needs to choose Σ with care. It is well known that, if the jump variances for each variable are too large, then we would expect most proposals to be either rejected, or even possibly, jump out of the support. If the variances are too small, then the acceptance rate increases, but we run the risk of not being able to explore the support adequately, or getting stuck in one of the disjoint regions centered on a local minimum.

EXAMPLE 4.6. Consider again the harmonic oscillator, and suppose $\epsilon = 2$. Recall that the acceptance region consists of three disjoint, closed regions, as in Figure 3.5. Suppose we set our starting point to be, for example, at the point $\theta' = (3.5, 3.5)$, which is in one of the regions surrounding a local minimum (but which is not the global minimum). We assume that the covariance matrix of the proposal distribution is $\frac{1}{\delta^2} I_2$, where I_2 is the 2×2 identity matrix. If we choose a very large δ, it is easy to see that, although our acceptance rate would be quite large, there is large probability of getting stuck in this "wrong" region. Again, as the acceptance rate will vary in between samples, we take the average when producing five separate samples. Table 4.1 shows that the average over five separate runs for the minimum $\rho(\theta)$ and acceptance rate as we vary δ where the starting point is θ'.

We can see in Table 4.1 that as the value of δ increases (and so the "jumps" become smaller), the acceptance rate also increases, which corresponds to our intuition. However, the minimum distance is increasing despite the larger number of accepted elements. This is because more and more values get stuck in the region which contains the local minimum.

TABLE 4.1. Results using MCMC, 1000 elements in the sample, prior distribution: uniform over $[0, 7] \times [0, 7]$. Average acceptance rate of 5 runs shown.

δ	Acceptance rate	Minimum distance	No. of runs with minimum > 1
1	50.4	0.0179	0
1.5	62.42	0.3298	1
2	69.4	0.6514	2
2.5	75	1.6198	5

There is no clear rule to choose δ, or more generally, to choose the covariance matrix of the proposal distribution. In specific cases, there are available results from MCMC theory to obtain the optimal acceptance rate. The most well-known result is that of Gelman, Roberts, and Gilks [**13**] who showed that for a target density of the form $\pi(x_1, x_2, ..., x_d) = f(x_1)f(x_2)...f(x_d)$ for some one-dimensional smooth density and proposal distributions of the form $N(0, \sigma^2 I_d)$, the optimal acceptance rate is about 0.44 when $d = 1$, and decreases to 0.234 for a d-dimensional target distribution where $d \to \infty$. This result was later shown to be true even for several other target densities. For example, Rosenthal and Roberts showed that the result still holds even for target densities of the form

$$\pi(x) = \prod_{i=1}^{d} C_i f(C_i x_i),$$

where the C_i's themselves are iid from some fixed distribution. For details on these and other developments on the optimal scaling of a random walk MCMC, one can refer to Section 4.2 in [**36**].

Burn-in

Burn-in refers to the practice of discarding a certain number (or percentage) of iterations at the start of an MCMC run. In theory, if the Markov chain is run for an infinite amount of time, then we are guaranteed that the distribution of the values in any trajectory of the Markov chain will converge to that of the stationary distribution. However, when one has a finite chain (which any MCMC run will necessarily produce) and the starting point is a region of low probability, the chain may not be able to spend enough time in the regions of higher probability to ensure that early points are not disproportionally represented in the resulting sample. This problem is especially important when computing estimates, such as the mean, from the resulting samples, as the mean is sensitive to outliers.

There is no fixed rule as to how many iterations are to be disregarded, and whether burn-in is even needed. For example, Gelman et. al. propose in [**11**] to burn-in the first half of the generated chain. However, they themselves also admit that discarding the early runs may not be the most efficient approach, as it decreases the size of the sample and may increase the error of estimation. Thus, it seems better just to choose a "good" starting point in an area which, we hope, is of high probability. Geyer [**14**] in fact argues that any trajectory started anywhere near the center of the stationary distribution does not require burn-in. He calls this practice harmless, but unncessary, and goes on to say that "any point you don't mind having in a sample is a good starting point."

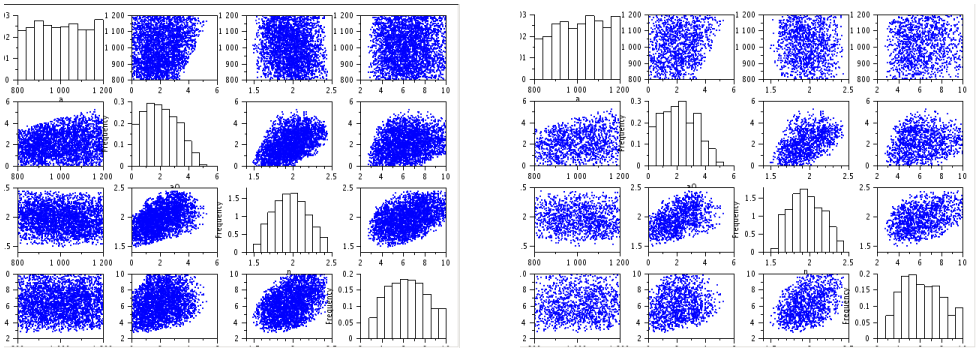
To test whether burn-in has any effect when estimating the coefficients in a differential system using an MCMC approach, we apply it to the repressilator

(see section 1.4.4). We examine the results of throwing away the first elements of the MCMC sample of size 5000, where the number of elements discarded ranges from 0 to 2500 in multiples of 500. To choose a starting point, we first run a rejection sample with 500 iterations, and use the coefficients that produce the minimum distance as our starting point, while we use $\delta = 0.25$ for the sample covariance matrix. Table 4.2 shows the effect of burn-in on the average distance for 5 runs of $\rho(\theta)$, where θ is either the minimum distance ($\hat{\theta}_{min}$) and the mean ($\hat{\theta}_{ave}$). We see that there is no advantage obtained by discarding the initial samples.

TABLE 4.2. Average value of $\rho(\hat{\theta})$ for 5 runs, where $\epsilon = 1000$, $\theta = (\alpha, \alpha_0, \gamma, \beta) = (1000, 1, 2, 5)$ are the "true" values of the coefficients, and the prior distribution is uniform over $[800, 1200] \times [0, 7] \times [0, 4] \times [0, 10]$.

No. of samples discarded	Average distance $\rho(\hat{\theta}_{min})$	$\rho(\hat{\theta}_{ave})$
0	9.834	281.972
500	9.834	287.196
1000	10.568	289.862
1500	14.886	279.866
2000	16.104	293.024
2500	17.388	315.98

In addition, Figure 4.2 below shows the effect of a burn-in phase in the acceptance rate and the resulting estimate. We see that other than the quantity of plotted points, there is no significant difference in the distribution of the resulting plots.



(A) Full sample (B) First half discarded

FIGURE 4.2. Scatter matrix of the MCMC sample based on the entire sample (a) and where the first half of the sample is discarded (b) in the example of the repressilator (see Section 1.4.4). Here, we assume that the "true" values for the coefficients are $\alpha_0 = 1, \gamma = 2, \beta = 5, \alpha = 1000$, and that we know the solution at $T = \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$.

4.1.4. Comparison between RS and MCMC. In this section, we compare the rejection method and the MCMC methods in estimating the coefficients of the repressilator once again. To do this, we perform simulation experiments with 5000 runs for both methods. To choose the starting point and the covariance matrix of the MCMC algorithm, we begin with 500 runs of the rejection method. Provided that we get a sufficient number of accepted samples, we shall choose the $\theta = (\alpha, \alpha_0, \gamma, \beta)$ that gives the minimum ρ as the starting point of the MCMC run, and 1/4 of the covariance matrix of the accepted samples as the covariance matrix of our Gaussian proposal distribution q.

TABLE 4.3. Results using RS and MCMC in the case of the repressilator, 1000 runs, prior distribution: uniform over $[0, 7] \times [0, 4] \times [0, 10] \times [800, 1200]$, average acceptance rate of 5 trials is shown.

ϵ	Rejection		MCMC	
	Min. distance	Acceptance rate	Min. distance	Acceptance rate
1200	27.73	8.384	11.488	58.34
1000	13.79	5.812	8.176	59.18
800	30.69	3.724	5.896	58.976
600	17.75	2.332	5.144	58.476
400	32.49	1.152	4.168	59.724

Table 4.3 provides the results of 5000 runs of both RS and MCMC methods as ϵ decreases. As already mentioned before, the acceptance rate of the rejection

method decreases as ϵ decreases. Thus, although each of the accepted samples should, in theory, have a smaller distance from θ_0, the minimum distance does not decrease in an analogous manner because the decreasing number of accepted samples is no longer able to cover enough of A_ϵ.

In contrast, the coefficients θ obtained using MCMC and the minimum distance metric become more and more accurate as ϵ decreases, as evidenced by the decreasing minimum ρ. This is due to the acceptance rate for MCMC remaining approximately fixed regardless of the value of ϵ. This acceptance rate is equal to

$$\tau = \int_{\theta \epsilon S} \int_{\theta' \epsilon S} \alpha(\theta, \theta') d\theta' d\theta,$$

where α is the acceptance probability defined in (4.1). This will be the case provided the starting point is chosen so that it is sufficiently close to the true value. In our case, we ensured that this occurs with large probability by choosing it to be the θ with the minimum distance, and requiring that the number of accepted samples in the preliminary sample is sufficiently large.

4.2. A Sequential Monte Carlo method

As we have already seen in the previous sections, one key factor in obtaining a good distribution and consequently a good estimate of the coefficients in a system of differential equations using our simulation-based method is obtaining a relatively large acceptance rate. In fact, one can really only have a good confidence in our results if we are able to produce a sufficiently large sample of points in a small neighborhood of the "best" coefficient θ_0. However, as the dimension of the coefficient space grows, it becomes increasingly difficult, or even impossible to choose a prior distribution that is centered well-enough on the good region in $\mathbb{R}^m$ to obtain a high acceptance rate. It then becomes even more important to have an efficient way to sample from the relevant parts of the support of our chosen prior distribution.

In this section, we shall examine a method based on Sequential Monte Carlo (SMC) ([**40**], [**44**]) which not only helps alleviate the aforementioned problem, but also that of getting stuck in local minima of S_0. Variants of this method are now widely used in many fields such as statistics, signal processing, and mathematical finance. We shall provide a straightforward presentation of a Sequential Monte Carlo method to sample from our target distribution $\pi_\epsilon(\theta|\bar{y})$ (recall equation (3.4)). After presenting the method in Section 4.2.1, we provide a short introduction to the theoretical basis of our SMC method in Section 4.2.2.

4.2.1. Presentation of the method. The method which we shall now present is based on that given by Toni et. al. in [**44**]. In this method, the objective is still to produce samples from our target distribution $\pi_\epsilon(\theta|\bar{y})$. Here,

the samples are usually called *populations* and any element of a sample a *particle*. However, to avoid having a problem with low acceptance rate associated with a small ϵ, we shall do this sequentially, beginning from a much higher ϵ_1 than our target threshold ϵ and gradually decreasing it until we reach ϵ.

To do this, we begin by choosing a prior distribution π_0 for θ and a sequence of decreasing thresholds $\epsilon_1 > \epsilon_2 > ... > \epsilon_S = \epsilon$. The number and choice of thresholds ϵ_i will not only define how many populations of samples the method will go through, but also how fast the law will converge towards our desired distribution. Next, we need to choose one Markov kernel $q_s(\theta, \theta^*)$ for each $s = 1, 2, ..., S$. For example, we can choose the Markov kernel to be multivariate Gaussian centered on the current element of the sample and with a fixed covariance matrix. To simplify matters, we can choose the same kernel for each s as we did for MCMC in the previous section. This kernel will determine how our particles will move around the coefficient space. Finally, we need to specify the number of *accepted* particles N_s we wish to have in each population. For simplicity, we will take $N_s = N$ for all s. If N is chosen large enough, we shall see later that the law of the population produced converges to our target distribution in a manner that will be made precise.

To obtain the first population of particles, one proceeds as in the rejection method presented in Section 3.3 using a (possibly high) threshold ϵ_1. This means that we shall generate one set of coefficients θ^* from $\pi_0(\theta)$, and compute a measure of distance $\rho(\theta^*)$ from θ^* to the known data $\bar{y}(T)$, where ρ is defined as in (3.3). As before, we keep θ^* only if $\rho(\theta^*) < \epsilon_1$. This is repeated until we obtain N accepted particles $\theta_1^{(1)}, \theta_2^{(1)}, ..., \theta_N^{(1)}$. We then assign specific *weights* $W_1^{(1)}, W_2^{(1)}, ..., W_N^{(1)}$ to each particle, which will be necessary for the proper convergence of the resulting empirical distribution defined by the $\theta_i^{(1)}$'s and $W_i^{(1)}$'s. For this first population, we assign equal weight to each particle, and so $W_i^{(1)} = 1/N$ for $i = 1, 2, ..., N$.

Suppose now that we have the particles $\theta_i^{(s-1)}$, $i = 1, 2, ..., N$ for the $(s-1)$st population, with corresponding weights $W_i^{(s-1)}$, $i = 1, 2, ..., N$. To obtain the particles $\theta_i^{(s)}$, $i = 1, 2, ..., N$ for the next population, we begin by taking a sample θ^* from $\theta_1^{(s-1)}, \theta_2^{(s-1)}, ..., \theta_N^{(s-1)}$, where the probability of selection is proportional to their previously computed weights $W_1^{(s-1)}, W_2^{(s-1)}, ..., W_N^{(s-1)}$. We then simulate a new particle θ^{**} from θ^* using the Markov kernel q_s, which we shall only keep if $\rho(\theta^{**}) < \epsilon_s$. If θ^{**} is accepted, we shall assign it a preliminary weight of

$$w_i^{(s)}(\theta^{**}) = \frac{\pi_0(\theta^{**})}{\sum_{i=1}^N W_i^{(s-1)} K_s(\theta_i^{(s-1)}, \theta^{**})}.$$

This process is repeated until we obtain N accepted samples. Once all N samples for population s have been generated, we normalize the preliminary weights. That is, we adjust the weight for each particle proportionally so that when taken together, the N particles sum up to 1:

$$W_i^{(s)} = \frac{w_i^{(s)}}{\sum_{i=1}^{N} w_i^s}.$$

The procedure which we have just described can then be repeated until we obtain the particles for population S, corresponding to our desired threshold $\epsilon_S = \epsilon$. The result of the method is a "particle estimate"

$$\hat{\pi}_s(\theta) = \sum_{i=1}^{N} W_i^{(s)} \delta_{\theta_i^{(s)}}(\theta)$$

of our target distribution $\pi_\epsilon(\theta|\rho(\theta) < \epsilon)$.

To give us an idea of how this SMC-based method works in practice, we now apply the method to a simple example.

EXAMPLE 4.7. We apply the sequential monte carlo method on the repressilator method, where the model coefficients are $\alpha_0 = 1, \gamma = 2, \beta = 5, \alpha = 1000$, and for the time points $T = \{0, 0.5, 1, ..., 3\}$. The sequence of thresholds is chosen as $\{1200, 800, 400\}$. The Markov kernel K_m for the 2nd and 3rd populations is assumed to be Gaussian centered on the selected value θ^* and with covariance matrix equal to 1/9 of the covariance of the 500 samples in the previous population. The results given in Table 4.4 are the average over 5 separate runs of the method.

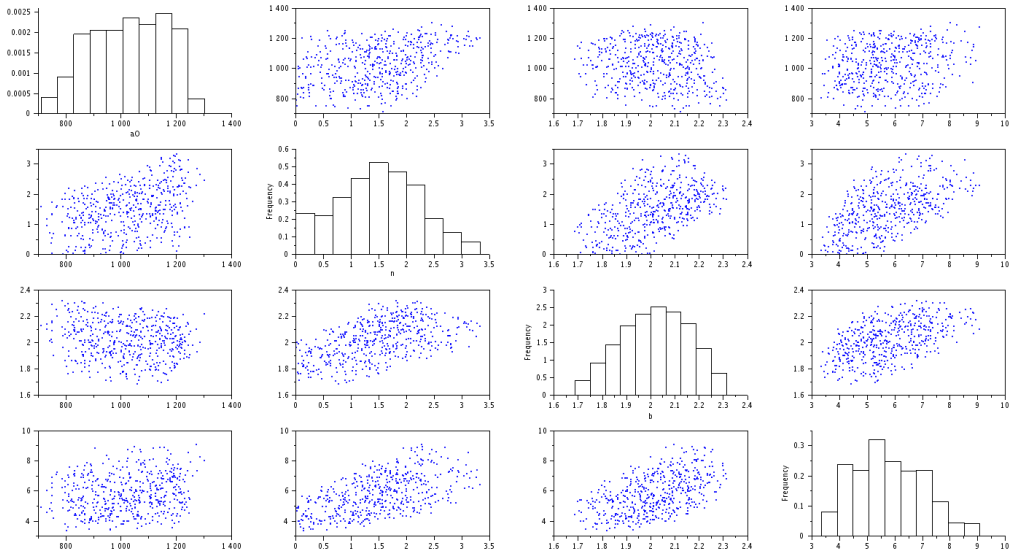
TABLE 4.4. Results of SMC for $\epsilon = \{1200, 800, 400\}$. True value: $\alpha = 1000, \alpha_0 = 1, \gamma = 2, \beta = 5$, prior distribution: uniform over $[800, 1200] \times [0, 4] \times [0, 7] \times [0, 10]$

ϵ	No. of runs	Acceptance rate	Minimum distance
1200	6062	8.25%	25.91
800	1175	42.56%	14.24
400	1987	25.17%	13.3

Comparing our results to those in Table 4.3, our acceptance in the first population is virtually the same as that of the rejection method. This was to be expected since the first population was effectively a rejection run with $\epsilon = 1200$. However, we see the improvement in the acceptance rate in the succeeding populations due to the local moves, with both being above 25% on the average. The minimum distance is not as good as that produced by MCMC. This is because of the number of accepted samples. Recall from Proposition 3.8 of the convergence in probability of minimum distance estimate. As we only limited

ourselves to 500 samples, the MCMC method with the larger number of runs will have a bigger chance of getting a "better" result. Figure 4.3 shows the scattermatrix of the accepted coefficients in the third population ($\epsilon = 400$), which is similar to the scattermatrices we obtained for MCMC in Figure 4.2.

FIGURE 4.3. Scattermatrix for the repressilator method using the sequential monte carlo method.



Thus, at first glance, it seems that SMC is clearly inferior to MCMC. However, there are several important points one needs to consider. In our MCMC method, we recall that there was a significant chance of getting trapped in local minima. Here, the chance of encountering this problem is substantially less. Also, the fact that we start with a substantially higher ϵ and gradually decrease to our desired target ϵ allows us to start at a much higher acceptance rate. This allows us to overcome the low acceptance rates in the initial phase of the rejection or MCMC method. This becomes especially helpful as the dimension of the coefficient space increases.

4.2.2. Some words about the general theory. Consider a density π on S_0, where $\pi : S_0 \to \mathbb{R}^+$ is known pointwise. Importance sampling is a general method to estimate properties of π or to obtain a "particle approximation" of functions of random variables with density π by using only samples from another distribution η. Of course, we assume that $\eta(\theta) > 0$ for (almost) all θ for which $\pi(\theta) > 0$. The distribution η is often called the *importance distribution* or the *instrumental distribution*.

Let φ be any measurable function. Importance sampling is based on a simple change of measure, given by the following identity:

$$\begin{aligned}\mathbb{E}_\pi(\varphi(\theta)) &= \int \varphi(\theta)\pi(\theta)d\theta \\ &= \int \varphi(\theta)\frac{\pi(\theta)}{\eta(\theta)}\eta(\theta)d\theta \\ &= \int \varphi(\theta)w(\theta)\eta(\theta)d\theta \\ &= \mathbb{E}_\eta(\varphi(\theta)w(\theta)),\end{aligned}$$

where

$$w(\theta) = \frac{\pi(\theta)}{\eta(\theta)} \tag{4.5}$$

and $\eta(\theta) > 0$ for almost all θ with $\varphi(\theta)\pi(\theta) \neq 0$. This means that instead of sampling from $\pi(\theta)$ directly, we can take an iid sample $\theta_1, \theta_2, ..., \theta_N$ from $\eta(\theta)$ instead, and correct the bias resulting from sampling from the wrong distribution by using the weight $w(\theta)$. It is assumed that we know the values of $\pi(\theta_i)$ and $\eta(\theta_i)$, and that it is easy to sample from η. This gives us the following empirical approximation of π:

$$\pi(\theta) = \sum_{i=1}^{N} w_i \delta_{\theta_i}(\theta)$$

where

$$w_i = \pi(\theta_i)/\eta(\theta_i). \tag{4.6}$$

An alternative method which can be even more useful in most cases is sequential importance sampling (SIS). In SIS, the target distribution $\pi = \pi_M$ is obtained through a series of intermediate distributions π_s, $s = 1, ..., S-1$. In each step, we use importance sampling to sample from each π_s using an importance distribution η_s that we will now define recursively. First, consider a Markov chain of kernel $q(\cdot, \cdot)$ on the support of π. At time $s = 1$, we can begin by choosing $\eta_1 = \pi_1$. Given η_{s-1}, the next importance distribution η_s is given by

$$\eta_s(\theta_s) = \int \eta_{s-1}(\theta_{s-1})K_s(\theta_{s-1}, \theta_s)d\theta_{s-1}, \tag{4.7}$$

We can then perform importance sampling using this proposal distribution. At each step s, the weights can be computed using the same formula as before (see 4.6), but using π_s and η_s instead of π and η. Since η_s cannot usually be computed pointwise, a typical solution is to approximate it by using the Monte Carlo estimate

$$\eta_{s-1}^N K_s(\theta_s) = \frac{1}{N}\sum_{i=1}^{N} K_s(\theta_{s-1}, \theta_s).$$

Thus, at the step s, the weighted discrete measure is

$$\widehat{\pi}_s(\theta) = \frac{1}{N}\sum_{i=1}^{N} W_i^{(s)} \delta_{\theta_i^{(s)}}(\theta),$$

where $W_i^{(s)} = w_i^{(s)} / \sum_{j=1}^{N} w_j^{(s)}$ is a "particle approximation" of π_s.

The main disadvantage with the SIS approach is the problem of *weight degeneracy*. This means that after just a few steps, most of the weight tends to become concentrated on a very small number of particles. This is undesirable as it wastes a large part of the computational time without exploring the entire support. To solve this problem, one can perform resampling. Instead of simply evolving the weights, we sample from the previous population at each step. The algorithm, which is often called Sequential Importance Resampling (SIR), is given as follows:

A1. Initialize the particles $\theta_i^{(1)} \in S_0$ by generating N independent samples with law π_0.
A2. Initialize the weights $W_i^{(1)}$ as in SIS.
A3. At each time step $j < S$, *resample* the population according to the current weights $W_i^{(j)}$, that is, for each $i = 1, 2, ..., N$ replace $\theta_i^{(j)}$ by $\theta^* = \theta_{I(i)}^{(j)}$, where $I(i)$ is a random index from $\{1, 2, ..., N\}$, selected with probabilities proportional to the weights $W_1^{(j)}, W_2^{(j)}, ..., W_N^{(j)}$.
A4. Simulate a new particle $\theta^{**} \in S_0$ from the particle θ^* according to the Markov kernel q_j. This will now become the value of $\theta_i^{(j+1)}$.
A5. Update the weights using the same formula as in SIS, and produce $W_{j+1}^{(i)}$.

The algorithm that we have introduced consists in building a sequence of populations

$$\{\theta_1^{(1)}, ..., \theta_N^{(1)}\}, \{\theta_1^{(2)}, ..., \theta_N^{(2)}\}, ..., \{\theta_1^{(S)}, ..., \theta_N^{(S)}\}$$

and weights

$$\{W_1^{(1)}, ..., W_N^{(1)}\}, \{W_1^{(2)}, ..., W_N^{(2)}\}, ..., \{W_1^{(S)}, ..., W_N^{(S)}\}.$$

The weights are defined by induction as follows:

$$W^{(1)}(\theta) = \frac{1}{N}$$

$$w^s(\theta) = \frac{\pi_0(\theta)\mathbb{1}_{\rho(\theta)<\epsilon_s}(\theta)}{\sum_{i=1}^{N} W_i^{(s-1)} q(\theta_i^{(s-1)}, \theta)}$$

$$W^{(s)}(\theta) = \frac{w^{(s)}(\theta)}{\sum_{i=1}^{N} w_i^{(s)}(\theta)}$$

The sth population is a sample of size N from the previous population $s-1$ with the law

$$\eta_s(\theta) = \sum_{i=1}^{N} W_i^{s-1} q(\theta_i^{s-1}, \theta) \mathbb{1}_{\rho(\theta)<\epsilon_s}$$

The last population obtained by this algorithm will have a law which is a good approximation of the target distribution $\pi_\epsilon(\theta|\bar{y})$ in the following sense:

PROPOSITION 4.8. *The SMC algorithm is such that the law of the last population* $\{\theta_1^{(S)}, \theta_2^{(S)}, ..., \theta_N^{(S)}\}$ *with weights* $\{W_1^{(S)}, W_2^{(S)}, ..., W_N^{(S)}\}$

$$\widehat{\pi}_S(\theta) = \sum_{i=1}^{N} W_i^{(S)} \delta_{\theta_i^{(S)}}(\theta) \tag{4.8}$$

converges when N *tends to infinity to* $\pi_\epsilon(\theta|\bar{y})$.

Conclusion

In this thesis, we have studied differential systems with random coefficients using a simulation approach. For the problem of computing the law of the solution at time $t^\star$ of a differential equation with random coefficients, we have seen that even in simplest cases, one will usually obtain a distribution where the pdf cannot be computed explicitly, and for which we need to rely on Monte Carlo simulation. However, we have also seen that this may not be effective in all cases. In the case of a Riccati equation where the solution explodes in finite time, displaying the histogram on a compact manifold using two charts is an effective way to draw the histogram. Another possibility would be to approximate the distribution using a polynomial chaos expansion.

For the question of computing a best distribution of the coefficients of a system of differential equations that fits a known trajectory, we have described the rejection sampling algorithm, which produces a distribution of points which have a high probability to be the true coefficients. This gave us the flexibility to not only take into consideration the errors and uncertainties in the known data, but at the same time, to still provide a point estimate if necessary. Assuming that a true value of the coefficients exists, we have seen through several examples that for low dimension problems and a small enough maximum threshold ϵ for the accepted coefficients, one can obtain a posterior distribution which allows us to compute good point estimates for the unknown coefficients θ. However, when either ϵ decreases or the number of coefficients increases, we have seen that percentage (and thus, number) of accepted elements decreases, and which generally results in less accurate estimates. These led us to sampling methods which somehow use the knowledge obtained in the first few iterations. Some possibilities which we have seen to be very effective in increasing the acceptance rate include the sequential rejection method, or using methods based on the Markov chain Monte Carlo and Sequential Monte Carlo algorithms.

Bibliography

[1] Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *J.R. Statistical Soc.* 72(3): 269-342.

[2] Cartan, H. (1967) *Calcul Différentiel*: Hermann, Paris.

[3] Comet, J.-P., Bernot, G., Das, A., Diener, F., Massot, Camille, and Cessieux, A. (2012). "Simplified Models for the Mammalian Circadian Clock." In *Proceedings of the 3rd International Conference on Computational Systems-Biology and Bioinformatics*, eds. Chan, J. H., Meechai, A., and Kwoh, C. K., 127-138. DOI: 10.1016/j.procs.2012.09.014

[4] Cosma, I. A. and Evers, L. (2010). *Markov chains and Monte carlo methods.*

[5] Courant, R. and Hilbert D. (1953). *Methods of Mathematical Physics.* John Wiley & Sons, New York.

[6] Del Moral, P. (2004). *Feynman-Kac formulae - Genealogical and Interacting Particle Systems with Applications*: Springer, New York.

[7] Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *J.R. Statistical Soc.* 68(3): 411-436.

[8] Doucet, A. and Johansen, A. M. (2011), "Particle filtering and smoothing: Fifteen years later." In Handbook of Nonlinear Filtering, eds. Crisan, D. and Rozovsky, B., Oxford University Press.

[9] Elowitz, M. B. and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature* 403: 335-338.

[10] Funaro, D. (1992). *Polynomial Approximation of Differential Equations.* Springer-Verlag, Berlin.

[11] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004) *Bayesian Data Analysis*, 2nd ed: Chapman & Hall, New York.

[12] Gelman, A., Bois, F.Y., and Jiang, J. (1996). Physiological pharmacokinetic analysis using population modeling and informative prior distributions. *Journal of the American Statisical Association* 91: 1400-1412.

[13] Gelman, A., Gilks, W. R., and Roberts, G. O. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability*, 7(1): 110-120.

[14] Geyer, C. (1996). "Introduction to Markov chain monte carlo." In *Markov Chain Monte Carlo in Practice*, ed. Gilks, W.R., Richardson, S., and Spiegelhalter, D. J., 59-74. Chapman & Hall, London.

[15] Ghanem, R. and Spanos, P. (1991). *Stochastic Finite Elements: A Spectral Approach*: Springer-Verlag, New York.

[16] Hurewicz, W. (1958). *Lectures on Ordinary Differential Equations*: The M.I.T. Press, Cambridge, U.S.A.

[17] Johnson, R. A. and Wichern, D. W. (2007). *Applied Multivariate Statistical Analysis*: Pearson Prentice Hall, U.S.A.

[18] Karr, A. F. (1993). *Probability*: Springer-Verlag, New York.

[19] Levy dit Vehel, P. (2014). *A Systematic Approach to Financial Model Validation*. Ph.D. Thesis. University of Nice-Sophia Antipolis: France.

[20] Liang, H. and Wu, H. (2008). Parameter Estimation for Differential Equation Models Using a Framework of Measurement Errors in Regression Models. *J. Am. Stat. Assoc.* 103(484): 1570-1583.

[21] Marjoram, P., Molitor, J., Plagnol, V., and Tavaré S. (2003). Markov Chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences* 100 (26): 15324-15328.

[22] McKinley, T., Cook, A. R., and Deardon, R. (2009). Inference in Epidemic Models without Likelihoods. *The International Journal of Biostatistics*, 5(1): 24.

[23] Marin, J.-M., Pudlo, P., Robert C. P., and Ryder, R. J. (2012). Approximate Bayesian Computational Methods. *Statistics and Computing* 22(6), 1167-1180.

[24] Marsaglia, G. (1964). Ratios of Normal Variables and Ratios of Sums of Uniform Variables. Mathematical note 348, Boeing Scientific Research Laboratories, USA.

[25] Marsaglia, G. (2006). Ratios of Normal Variables. *Journal of Statistical Software* 16(4): 1-10.

[26] Marsaglia, George (1972). Choosing a Point from the Surface of a Sphere. *The Annals of Mathematical Statistics* 43(2), 645-646. doi:10.1214/aoms/1177692644. http://projecteuclid.org/euclid.aoms/1177692644.

[27] Meyn, S. P. and Tweedie, R. L. (2008). *Markov Chains and Stochastic Stability*, 2nd ed: Cambridge University Press, U.K.

[28] Nocedal, J. and Wright, S.J. (1999). *Numerical Optimization*: Springer-Verlag, New York.

[29] Oladyshkin, S. and Nowak, W. (2012). Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering & System Safety*, 106: 179190, 2012. DOI: 10.1016/j.ress.2012.05.002.

[30] Osborne, M. R. (2008). The Bock iteration for the ODE estimation problem. http://maths-people.anu.edu.au/~mike/Bock60.pdf, retrieved April 4, 2014.

[31] Pardoux, E. (2008). Markov Processes and Applications: Algorithms, Networks, Genome and Finance: John Wiley & Sons, Ltd, Chichester, UK.

[32] Pritchard, J., Seielstad, M., Perez-Lezaun, A., Feldman, M. (1999). Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution* 16: 1791-1798.

[33] Ramsay, R., Hooker, G., Campbell, D., Cao, J. (2007) Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society* 69(5): 741-796.

[34] Robert, C. and Casella, G. (2004). *Monte Carlo Statistical Methods*, 2nd ed: Springer-Verlag, New York.

[35] Robert, C. (1991). Generalized inverse normal distributions. *Statistics & Probability Letters* 11: 37-41.

[36] Rosenthal, J. (2011). "Optimal Proposal Distributions and Adaptive MCMC." In *Handbook of Monte Carlo*, ed. Brooks, A., Gelman, A., Jones, G.L., Meng X.L., 93-111. Chapman & Hall, New York.

[37] Rubin, D. B. (1984). Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician. *The Annals of Statistics* 12: 1151-1172.

[38] Scilab help: Optim - Nonlinear optimization routine. https://help.scilab.org/doc/5.3.3/en_US/optim.html. Retrieved March 7, 2014.

[39] Baudin, M., Couvert, V., and Steer, S. Optimization in scilab. https://www.scilab.org/content/download/.../optimization_in_scilab.pdf. Scilab consortium, retrieved March 12, 2014.

[40] Sisson S. A., Fan, Y., and Tanaka, M. (2007). Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences* 104(6): 1760-1765.

[41] Sisson S. A., and Fan, Y. (2011). "Likelihood-Free MCMC." In *Handbook of Monte Carlo*, ed. Brooks, A., Gelman, A., Jones, G.L., Meng X.L., 313-335. Chapman & Hall, New York.

[42] Stanescu, D., and Chen-Charpentier, B.M. (2008). Random coefficient differential equation models for bacterial growth. *Mathematical and Computer Modelling* 50(2009), 885-895.

[43] Tierney, L. (1996). "Introduction to general Markov chain theory." In *Markov Chain Monte Carlo in Practice*, ed. Gilks, W.R., Richardson, S., and Spiegelhalter, D. J., 59-74. Chapman & Hall, London.

[44] Toni, T., Welch, D., Strelkowa, N., Ipsen, A., Stumpf, M. (2009). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface* 6(31): 187-202.

[45] Varah, J. M. (1982). A Spline Least Squares Method for Numerical Parameter Estimation in Differential Equations. *SIAM J. Sci. and Stat. Comput.* 3(1): 2846.

[46] Vershynin R. (2012). Approximating the moments of marginals of high dimensional distributions. *Journal of Theoretical Probability* 25: 655-686.

[47] Williams, D. (1991). *Probability with Martingales*: Cambridge University Press, U.S.A.

[48] Xiu, D., and Karniadakis, G.E. (2002). The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations. *SIAM J. Sci. Comput.* 24(2): 619-644.

[49] Xiu, D. (2010). *Numerical Methods for Stochastic Computations: A spectral method approach*: Princeton University Press, New Jersey, U.S.A.

APPENDIX A

Scilab code

In this appendix, we provide the scilab code used to run the rejection, MCMC, and SMC methods in Chapters 3 and 4. We also include the code used to generate some of the figures in the entire thesis. For a more comprehensive collection of the source code in this work, one may refer to the following website: http://math.unice.fr/~chanshio

The following program, rs-mcmc-repressilator.sce, produces a sample of accepted parameters using the rejection method described in Chapter 3, and the MCMC method described in Section 4.1. One can easily switch between the methods by choosing the value of the variable mode, with mode=1 corresponding to rejection and mode=2 corresponding to MCMC.

```
//////////////////////////////////////////////////////////////////////
// This program implements the rejection and MCMC method to estimate the
// parameters in a repressilator model.
//////////////////////////////////////////////////////////////////////

clear;
funcprot(0);

time1=getdate('s');

for im=1:1
// im=1;
// parameters for the "known" trajectory
minval=0;           // lower bound of interval
maxval=7;          // upper bound of interval
numintervals=8;   // number of points

accept=0;  // just a counter for the number of accepted samples
delta=1000;

// true values of a and b (to be plotted in graph)
true_a=1000; true_a0=1; true_n=2; true_b=5;
// value of a and b in the ODE; equal to the true ones initially to generate
// the known trajectory, but this will be changed for each step.
a=true_a; b=true_b; n=true_n; a0=true_a0;

// number of runs
```

```
numsamples = 2500;

// which method? (1 = rejection, 2 = MCMC)
method = 2;
// use burn-in or not? (0 or 1)
burnin = 0;
// index of the first accepted sample in MCMC
firstaccept = 1;

// starting guess (only for MCMC)
start_a=grand(1,1,'unf',800,1200);
start_a0=grand(1,1,'unf',0,2);
start_n=grand(1,1,'unf',0,6);
start_b=grand(1,1,'unf',0,10);

// initial point of the differential system
t0=0;
u0=[0;2;0;1;0;3];
t=linspace(minval,maxval,numintervals);

// defines the repressilator ODE
function dy=g(t,u)
   dy(1) = -u(1)+a/(1+(u(6))^n)+a0;
   dy(2) = -b*(u(2)-u(1));
   dy(3) = -u(3)+a/(1+(u(2))^n)+a0;
   dy(4) = -b*(u(4)-u(3));
   dy(5) = -u(5)+a/(1+(u(4))^n)+a0;
   dy(6) = -b*(u(6)-u(5));
endfunction

// get the sum of squares difference
function mydiff=getSSdiff(x,y)
   mydiff = delta + 1;
   if size(x) == size(y) then
       diff=(x-y);
       mydiff = sum(diff .* diff);
   end;
endfunction

// Multivariate gaussian pdf
function val=mnpdf(x,mean,sigma)
   k = size(mean,"r");
   val=abs(det((2*%pi)^k * sigma))^(-0.5)*exp(-0.5*(x-mean)' *inv(sigma)*(x-mean));
endfunction

y=ode(u0,t0,t,g);

mysample = zeros(4,numsamples+1); // initialize vector of samples
if method==2 then
   mysample(:,1)=[900;1.5;1.5;7.5]; // chosen starting guess (expect to get 1,1)
end
```

```
myls = zeros(1,numsamples); // vector of differences

covmat = [10000,0,0,0; 0,0.25,0,0; 0,0,2.25,0; 0,0,0,6.25]; // covariance matrix (only
     for MCMC)

for i=1:numsamples
    // method 1: rejection
    if method == 1 then
        a=grand(1,1,'unf',800,1200);
        a0=grand(1,1,'unf',0,2);
        n=grand(1,1,'unf',0,6);
        b=grand(1,1,'unf',0,10);

        params=[a,a0,n,b];

        mysample(:,i+1) = mysample(:,i)
        proposed=ode(u0,t0,t,g);

        // current metric: sum of squared differences
        ls=getSSdiff(proposed,y);

        myls(i) = ls;

        if ls <= delta then
           accept = accept + 1;
           mysample(:,accept) = [a;a0;n;b];
           //myls(accept) = ls;
        end
        if modulo(i,2000)==0 then
           mprintf("Current run: %i \n", i);
           //mprintf("Number of accepted values: %i \n",accept);
        end
    else
        // start MCMC
        d=grand(1,'mn',mysample(:,i),covmat);
        a=d(1); a0=d(2); n=d(3); b=d(4);
        //end;

        mysample(:,i+1) = mysample(:,i)
        // Compute the first sum of squares distance. This is necessary so that there
        // will be a distance stored for the first sample in case it is rejected
        if i == 1 then
           myls(i) = getSSdiff(ode(u0,t0,t,g),y);
        else
        // ensures vector of differences has similar indices as vector of samples
           myls(i) = myls(i-1);
        end;

        if (a<=1200)&(a>=800)&(a0<=2)&(a0>=0)&(n<=6)&(n>=0)&(b<=10)&(b>=0) then
            proposed=ode(u0,t0,t,g);
```

```
            // current metric: sum of squared differences
            ls=getSSdiff(proposed,y);
            if i == 1 then
                myls(i) = ls;
            end

            if ls <= delta then
                u = rand(1,1);
                num = mnpdf(d,mysample(:,i),covmat);
                den = mnpdf(mysample(:,i),d,covmat);

                if u <= num/den then
                    if accept == 0 then
                        firstaccept = i;
                    end
                    mysample(:,i+1) = [a;a0;n;b];
                    accept = accept + 1;
                    myls(i) = ls;
                end
            end
        end
    end
end

mprintf("\nRun No.: %i",im);
mprintf("\nNumber of accepted values: %i \n", accept);

if method==1 then
    // Rejection
    [p,q] = min(myls(1:accept));
    mysample = mysample(:,1:accept);
else
    // MCMC
    if burnin == 1 then
        myls = myls(:,firstaccept:numsamples);
        mysample = mysample(:,firstaccept:numsamples);
    end
    [p,q] = min(myls);
end

mmean = mean(mysample,'c');
mstdev = stdev(mysample,'c');

if method==2 then
    // +1, because the first position is occupied by the initial point
    mprintf("Best guess: a=%f a0=%f n=%f b=%f \n",
         mysample(1,q+1),mysample(2,q+1),mysample(3,q+1),mysample(4,q+1));
else
    mprintf("Best guess: a=%f a0=%f n=%f b=%f \n",
         mysample(1,q),mysample(2,q),mysample(3,q),mysample(4,q));
```

```
end

mprintf("Mean of accepted values: a=%f, a0=%f, n=%f, b=%f \n",
     mmean(1),mmean(2),mmean(3),mmean(4));
mprintf("SD of accepted values, a=%f, a0=%f, n=%f, b=%f \n", mstdev(1), mstdev(2),
     mstdev(3), mstdev(4));
mprintf("\nTime elapsed: %i seconds \n", getdate('s') - time1);
//end


s = size(mysample);
if s < 0 then
    xset("window",0);
    clf();
    // plot all the accepted points (initial point plot bug not yet fixed)
    plot(mysample(1,:),mysample(2,:),'x');
    // plot the "true" values of a and b
    plot(true_a,true_b,'.r');
    mtlb_axis([minunf,maxunf,minunf,maxunf]);
end

end
```

The following program, smc-repressilator.sce, estimates the parameters in a repressilator model using the SMC method described in Section 4.2.

```
////////////////////////////////////////////////////////////////////////
// This program implements the SMC method to estimate the parameters in
// a repressilator model.
////////////////////////////////////////////////////////////////////////
clear;
funcprot(0);
// to track total run time.
starttime = getdate('s');
rand('seed',starttime);

// parameters for the time the known data are given
minval=0;           // lower bound of interval
maxval=7;          // upper bound of interval
numintervals=8;   // number of division points of interval

popcounter = 0;
// number of accepted samples needed per population
numsamples = 500;

// thresholds for sum-of-squares error
deltas=[5000;3000;2000;1000];
z=size(deltas);
populations = z(1);

// the variable "runcounter" below records the number of runs needed to generate the
```

```
// required number of accepted samples (indicated by "numsamples" above)
runcounter = zeros(1,4);

// "true values" of the parameters
a=1000; a0=1; n=2; b=5;
// initial value of the ODE
u0 = [0;2;0;1;0;3];
t0 = 0;
// time points of the model data
t=linspace(minval,maxval,numintervals);

// define the repressilator ODE
function dy=g(t,u)
   dy(1) = -u(1)+a/(1+(u(6))^n)+a0;
   dy(2) = -b*(u(2)-u(1));
   dy(3) = -u(3)+a/(1+(u(2))^n)+a0;
   dy(4) = -b*(u(4)-u(3));
   dy(5) = -u(5)+a/(1+(u(4))^n)+a0;
   dy(6) = -b*(u(6)-u(5));
endfunction

// return the value of the multivariate normal pdf
function val=mnpdf(x,mean,sigma)
   k = size(mean,"r");
   val=abs(det((2*%pi)^k * sigma))^(-0.5)*exp(-0.5*(x-mean)'*inv(sigma)*(x-mean))
endfunction

// computes the distance via the sum of squared differences
function mydiff=getSSdiff(x,y)
   mydiff = deltas(popcounter+1)+1;
   if size(x) == size(y) then
       diff=(x-y);
       mydiff = sum(diff .* diff);
   end;
endfunction

// get the index in the array arr which contains the largest value <= th (using binary
    search)
function ind = getmaxindex(arr,th)
    ind = 1;
    found = 0;

    mid = ceil(length(arr)/2);
    bottom = 1; top = length(arr);

    while (found <> 1) // just to prevent infinite loops
        if arr(mid) <= th then
            if  mid == length(arr) then
               found = 1;
               ind = mid;
            elseif (arr(mid+1) > th) then
```

```
            found = 1;
            ind = mid+1;
        else
            bottom = mid;
            mid = ceil((mid + top)/2);
        end
    else
        if mid == bottom then
            found = 1;
            ind = bottom;
        end
        top = mid;
        mid = floor((bottom + mid)/2);
    end
  end;
endfunction

y=ode(u0,t0,t,g);//[1.0474206 1.7477874 1.4509635 0.6791915 0.1743489 0.6147745
    0.2994606 1.5345684; 0.4437047 0.7159919 1.3118426 1.9396839 1.3564853 0.3901931
    0.7139732 0.4825672]
//y=ode(u0,t0,t,g)+myrand; // assumed observed data for comparison

// Vector of particles (each row is one population)
// In reality, we don't have to store all the intermediate populations, but they will
// be useful for looking at the evolution of the estimated distribution.
mysamplea = zeros(populations,numsamples);
mysamplea0 = zeros(populations,numsamples);
mysamplen = zeros(populations,numsamples);
mysampleb = zeros(populations,numsamples);
// vector of weights
weights = ones(populations,numsamples);
// just to assess if the code runs properly
randval = zeros(1,numsamples);
myls = zeros(numsamples);

// Main loop to compute the accepted parameters
while (popcounter < populations)
  w = 0;
  i = 1;
  runcount = 0;

  while i <= numsamples
    if popcounter > 0 & i == 1 then
      aa =
          [mysamplea(popcounter,:);mysamplea0(popcounter,:);mysamplen(popcounter,:);mysampleb(popcounter,:)]
      psd = 1/9*cov(aa');
    end;
   // First population: just a typical ABC-rejection sample from the prior distribution
    if popcounter == 0 then
      // a = grand(1,1,'nor',mu,sd);
      a = grand(1,1,'unf',800,1200);
```

```
        a0 = grand(1,1,'unf',0,7);
        n = grand(1,1,'unf',0,4);
        b = grand(1,1,'unf',0,10);
    else
// Second population and later: Choose one of the particles
// in the previous population at random, based on their weights.
        // take a random value
        randval(i) = rand();
        // find which sample corresponds to this random number.
        j = getmaxindex(cum,randval(i));
        current_value =
             [mysamplea(popcounter,j);mysamplea0(popcounter,j);mysamplen(popcounter,j);mysampleb(popcounter,j)]
        // take a Gaussian sample centered on current_value
        sampled = grand(1,'mn',current_value,psd);
        // these are now our proposed parameter values.
        a=sampled(1); a0=sampled(2); n=sampled(3); b=sampled(4);
    end

    // Simulate a candidate dataset using the sampled parameter
    cand=ode(u0,t0,t,g);

    // metric: sum of squared differences
    ls = getSSdiff(cand,y);
    // if accepted, we store the result, including the distance
    if ls <= deltas(popcounter+1) then
        myls(popcounter+1,i) = ls;
        mysamplea(popcounter+1,i) = a;
        mysamplea0(popcounter+1,i) = a0;
        mysamplen(popcounter+1,i) = n;
        mysampleb(popcounter+1,i) = b;
        i = i + 1;
    end

    runcount = runcount+1;
end
// normalize the weights

//if popcounter == 1 & i=1 then
//    psd = cov(mysample');
//end

// after the first population, the computation of the weights becomes more
     complicated.
if popcounter >=1 then
    for k=1:numsamples
        den = 0;
        // compute the numerator of the weight - currently, it's uniform
        num = 1/(400*7*4*10);
        // sum for denominator
        for l=1:numsamples
```

```
                old =
                    [mysamplea(popcounter+1,l);mysamplea0(popcounter+1,l);mysamplen(popcounter+1,l);mysampleb(popcou
                new =
                    [mysamplea(popcounter+1,k);mysamplea0(popcounter+1,k);mysamplen(popcounter+1,k);mysampleb(popcou
                den = den + weights(popcounter,l)*mnpdf(old,new,psd);
            end

            weights(popcounter+1,k)=num/den;
        end
    end
    // normalize the weights
    weights(popcounter+1,:) = weights(popcounter+1,:)/sum(weights(popcounter+1,:));

    // Display the results of the current population
    endtime = getdate('s');
    mprintf("Population %i complete, total time elapsed now: %i seconds \n",
        popcounter+1, endtime-starttime);
    mprintf("Number of trials: %i \n", runcount);
    [p,q] = min(myls(popcounter+1,:));
    // Compute and display the distance of the "best guess" parameters from the known
        data.
    a=mysamplea(popcounter+1,q); a0=mysamplea0(popcounter+1,q);
    n=mysamplen(popcounter+1,q); b=mysampleb(popcounter+1,q);
    mprintf("Minimum distance: %f \n", getSSdiff(y,ode(u0,t0,t,g)));

    // prepare parameters for the next population, if any.
    starttime = endtime;
    popcounter = popcounter + 1;
    cum = cumsum(weights(popcounter,:));
    runcounter(1,popcounter) = runcount;
end
```

This program generates the histograms in Figure 2.7 at time $maxval = 5$ of the linear equation $y' = -A * y + 1$ and its corresponding Riccati equation $z' = -z^2 + Az$ where $A \sim N(1, 4)$, and using the change of manifold technique strategy discussed in section 2.2.

```
////////////////////////////////////////////////////////////////////////////
// This program computes the histogram at time T=maxval of a logistic ODE with random r
// using two "charts" to avoid the problem of poles in finite time of the logistic ODE.
////////////////////////////////////////////////////////////////////////////

clear;
funcprot(0)

// parameters of the Gaussian distribution.
// for r
mu1=1;
sd1=2;
// for K (K is fixed, and equal to 1 at the moment)
```

```
mu2=1;
sd2=0;

//minval=0           // lower bound of interval
maxval=5          // upper bound of interval
//interval=(maxval-minval)/100 // distance between consecutive data points
numofruns=1000;    // number of samples.

// generate a sample from the normal distribution
myrand=grand(numofruns,1,'nor',mu1,sd1);
//myrand2=grand(numofruns,1,'nor',mu2,sd2);

seuil = 5;
bas = -5;

// initial parameters of the ODE. With our logistic ODE with K=1, we will encounter
// the problem of pole in finite time if the sampled r is positive.
t0=0;

//t=minval:interval:maxval;

z = zeros(1,numofruns);

for i=1:numofruns
    function udot=g1(t,u)
        udot = -myrand(i)*u+1;
    endfunction

    function udot=g2(t,u)
        udot = -u^2+myrand(i)*u;
    endfunction

    startpoint = -1;

    // Stay in logistic
    //if (myrand(i)<0 & startpoint > 1/myrand(i)) then
    //if ((myrand(i)<0 & startpoint > 1) | (myrand(i)> 0 & startpoint < 0))
    if (myrand(i)>startpoint) then
        u0 = startpoint;
        z(i) = ode(u0,t0,maxval,g1);
        //u0=-2;
    else
        // Convert to linear
        //u0=-0.5;
        u0 = 1/startpoint;
        z(i) = 1 ./ ode(u0,t0,maxval,g2);
    end
end

z1 = z;
for i=1:numofruns
```

```
    if z1(i) > seuil then z1(i) = seuil;
    else if z1(i) < bas then z1(i) = bas;
        end
    end
end

clf();

y1 = linspace(0.01,5.01,100);

a1 = exp(-((y1-0)^(-1)-mu1)^2 ./(2*sd1^2));
b1 =(sqrt(2*%pi)*(y1-0)^2*sd1)^(-1);
c1 = a1.*b1;

xset("window",0);
clf();
histplot(100,z1);
//plot(y1,c1);
mtlb_axis([-1,1,0,1.5]);
//mtlb_axis([-3,3,0,1]);
xtitle("","Y","f(Y)");
axes = gca();
//axes.auto_ticks = ["off","off","off"];
//axes.x_ticks = tlist(["ticks", "locations","labels"],.. // continuation in next line
//    [0 1 2 3 4 5], ["0", "1", "2", "3","4", "5"]);
//axes.y_ticks = tlist(["ticks", "locations","labels"],.. // continuation in next line
//    [0 0.2 0.4 0.6 0.8 1 1.2], ["0","0.2","0.4","0.6","0.8","1"]);
    //[0 0.1 0.2 0.3 0.4 0.5], ["0", "0.1", "0.2", "0.3","0.4", "0.5"]);
//axes.sub_ticks=[3,4];
axes.font_size=3;
axes.x_label.font_size=3;
axes.y_label.font_size=3;

z2 = 1 ./z;
for i=1:numofruns
    if z2(i) > seuil then z2(i) = seuil;
    else if z2(i) < bas then z2(i) = bas;
        end
    end
end

a2 = exp(-(y1-mu1) .^2 ./(2*sd1^2));
b2 =(sqrt(2*%pi)*sd1) .^(-1);
c2 = a2.*b2;

xset("window",1);
clf();
histplot(100,z2);
//plot(y1,c2);
mtlb_axis([-1,1,0,3]);
xtitle("","Z","f(Z)");
```

```
axes = gca();
axes.font_size=3;
axes.x_label.font_size=3;
axes.y_label.font_size=3;
```

The following code produces the contour plots of the distance function $\rho(\theta)$ for the harmonic oscillator $x' = -ay, y' = bx$ in Figure 3.4a.

```
////////////////////////////////////////////////////////////////////////////
// This program draws the contour plot of the distance function Rho of the solution
// of the harmonic oscillator to the known data. This is done by computing Rho
// over a grid of test parameters within the support of the prior distribution.
////////////////////////////////////////////////////////////////////////////

clear;
clf();
funcprot(0);

// Time points where we have known data. We assume the intervals are regular.
minval=0;           // lower bound of interval
maxval=3;           // upper bound of interval
numintervals=7;     // number of points (for the linspace)

// true values of the model parameters a and b

true_a=1.5; true_b=1.5;

// value of a and b in the ODE; equal to the true ones initially to generate
// the known trajectory, but this will be changed for each step.

a=true_a; b=true_b;

// initialize the differential system
t0=0;
u0=[1;0.5];

t=linspace(minval,maxval,numintervals);

// initialize the grid of points. Here, we assume that the support is [0,3]x[0,3], and
     we have a 101x101 grid of points in the support for which we will compute the value
     of Rho.

xx = linspace(0,3,101);
yy = linspace(0,3,101);

// setup harmonic oscillator differential system

function dy=g(t,u)
   dy(1) = -a*u(2);
   dy(2) = b*u(1);
endfunction
```

```
// get the sum of squares difference / value of Rho
function mydiff=getSSdiff(x)
   diff=(x-y);
   mydiff = sum(diff .* diff);
endfunction

// generate the "reference trajectory" / known data
y=ode(u0,t0,t,g);

// produce a vector myls which will store all the Rhos
sizea = prod(size(xx));
sizeb = prod(size(yy));
myls = zeros(sizea,sizeb);

// this loop computes the value of Rho for each test parameter value
for i=1:sizea
    for j = 1:sizeb
        a=xx(i); b=yy(j);
        proposed=ode(u0,t0,t,g);
        myls(i,j)=getSSdiff(proposed);
    end
end

// construct the contour map.
contour2d(xx,yy,myls,linspace(0,5,11));

// change thickness of one particular contour. Here we chose the 5th from the largest,
     or where epsilon = 3.
tmp = gce();
curve = tmp.children;
curve(5).children.thickness = 3;
```

The following program, data.sce, produces the perturbed data which was used in the Section 3.6. It gives the user three options:

(1) Add a small Gaussian error to each component of the solution of the reference trajectory,

(2) Change the reference parameters when producing each point in the perturbed trajectory.

The result is two sets of perturbed data, which is found in the arrays $y1$ and $y2$, and the corresponding plots of these points. In this case, we used the competing species model (3.7).

```
////////////////////////////////////////////////////////////////////////////////
// This program was used to produce data using the two types of noise which were
// discussed in the first part of Section 3.6 in the text.
////////////////////////////////////////////////////////////////////////////////

funcprot(0);
```

```
// specify the times where the data is known
minval=0;
maxval=3;
numintervals=7;
t=linspace(minval,maxval,numintervals);

// reference values of the parameters
a=1; b=1.5;
new_a=a; new_b=b;

// initial point of the differential system
t0=0;
u0=[1;0.5];

function dy=g1(t,u)
   dy(1) = a*u(1) - (u(1))^2 - 0.5*u(1)*u(2);
   dy(2) = b*u(2) - 0.5*(u(2))^2 - 1.5*u(1)*u(2);
endfunction

function dy=g2(t,u)
   dy(1) = new_a*u(1) - (u(1))^2 - 0.5*u(1)*u(2);
   dy(2) = new_b*u(2) - 0.5*(u(2))^2 - 1.5*u(1)*u(2);
endfunction

// first option: add a small Gaussian noise to each entry.
myrand1=grand(2,numintervals,'nor',0,0.1);
y1=ode(u0,t0,t,g1)+myrand1;
xset("window",0);
clf();
plot(y1(1,:),y1(2,:));

// second option: change the parameters randomly in between each two times.
y2=zeros(2,numintervals);
y2(:,1)=u0;
for i=2:numintervals
    new = grand(1,'mn',[a;b],[0.01,0;0,0.01]);
    new_a = new(1);
    new_b = new(2);
    y2(:,i) = ode(y2(:,i-1),t(i-1),t(i),g2);
end
xset("window",1);
clf();
plot(y2(1,:),y2(2,:));
```