Bayesian Inference of Stochastic Degradation Models: A Likelihood-Free Approach

by

Indranil Hazra

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Civil Engineering

Waterloo, Ontario, Canada, 2021

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Dr. Faisal Khan
Professor, Dept. of Process Engineering
Memorial University of Newfoundland

Supervisor: Dr. Mahesh D. Pandey
Professor, Dept. of Civil & Environmental Engineering
University of Waterloo

Internal Members: Dr. Sriram Narasimhan
Adjunct Professor, Dept. of Civil & Environmental Engineering
University of Waterloo

Dr. Wei-Chau Xie
Professor, Dept. of Civil & Environmental Engineering
University of Waterloo

Internal-External Member: Dr. Kumaraswamy Ponnambalam
Professor, Dept. of Systems Design Engineering
University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The structural integrity and system performance of large engineering systems are adversely affected by various forms of degradation mechanisms. Modeling of such mechanisms is accomplished by collecting degradation data from periodic in-service inspections of structures and components. Subsequently, the degradation prediction is transformed into system and component lifetimes that are necessary inputs into the risk-based life-cycle management of critical structures. Stochastic degradation models are widely applicable for predicting degradation growths in structural components. The statistical estimation of such models is often challenged by various uncertainties, such as inherent randomness of a degradation process, parameter uncertainty due to noise in measurements, coverage issues, probe signal loss, the limited resolution of the inspection probe, and small sample size.

The Bayesian inference method can be used to quantify the uncertainties of the model parameters. However, degradation data of engineering structures are often contaminated by a significant amount of inspection errors added by various inspection tools. As a result, the likelihood function becomes analytically intractable and computationally expensive to a degree that any traditional likelihood-based Bayesian inference scheme (e.g., Gibbs Sampler, Metropolis-Hastings sampler) turns difficult for practical use.

This study proposes a practical likelihood-free approach for parameter estimation based on the approximate Bayesian computation (ABC) method. ABC is a simulation-based approach that does not require an explicit formulation of the likelihood function. Three advanced computational algorithms, namely, ABC using Markov chain Monte Carlo (ABC-MCMC), ABC using Hamiltonian Monte Carlo (ABC-HMC), and ABC using subset simulation (ABC-SS), are developed and implemented for the parameter estimation task. In the context of degradation modeling, various implementation issues

of these algorithms are discussed in detail.

To improve the mixing properties of ABC-MCMC, a new ABC algorithm is derived based on the HMC sampler that uses the Hamiltonian dynamics to simulate new samples from its seed samples. Its non-random walk behavior helps to explore the target probability space more effectively and efficiently than the standard random-walk MCMC method. The convergence of the proposed ABC-HMC algorithm is proved by satisfying the detailed balance equation, and its efficacy is verified using a numerical example. Furthermore, A new sequential ABC algorithm is proposed to deal with highly diffused priors in a Bayesian inference problem. The proposed ABC algorithm is based on the subset simulation method and a modified HMC algorithm. With faster convergence, the new algorithm turns out to be a powerful method to sample from a complex multi-modal target density as shown by a numerical example. The applicability of the proposed algorithm is further extended by transforming it into a likelihood-free Bayesian model selection tool.

The proposed likelihood-free approach for Bayesian inference is applied to analyze practical data sets from the Canadian nuclear power plants. The practical data consist of two types of degradation measurements: (1) wall thickness data of the feeder pipes that are affected by the flow-accelerated corrosion (FAC) and (2) data from the steam generator tubes affected by the pitting corrosion. Four popular stochastic degradation models are considered, namely, the random rate model, the gamma process model, the mixed-effects regression model, and the Poisson process model, for characterizing the degradation processes under study. In the modeling process, various inspection uncertainties, such as the sizing error, the coverage error, and the probability of detection (POD) error are taken into account. The numerical results demonstrate that, in comparison to the likelihood-based approach, the proposed likelihood-free approach notably reduces computational time while accurately estimating the model parameters. This

study finds that these intuitive and easy-to-implement likelihood-free algorithms are versatile tools for Bayesian inference of stochastic degradation models and a promising alternative to the traditional Bayesian estimation methods.

Acknowledgments

I would like to express my deepest gratitude to my wonderful supervisor Prof. Mahesh D. Pandey for providing me the opportunity to pursue doctoral studies at the University of Waterloo, and for his constant support, guidance, and encouragement to carry out independent research throughout my doctoral studies. This thesis could not have been completed without his help.

I am grateful to my Ph.D. defense committee members Prof. Faisal Khan, Prof. Sriram Narasimhan, Prof. Wei-Chau Xie, and Prof. Kumaraswamy Ponnambalam for agreeing to review my thesis and give valuable suggestions. I would like to thank Prof. Sriram Narasimhan for always being very kind and supportive. Special thanks go to Mikko I. Jyrkama and Noldainerick Manzana for their valuable input. I owe my deepest gratitude to Rajdip Nayek, who has been a great mentor since the beginning of my Ph.D. journey. I would also like to thank Jane Russwurm and Mary McPherson for helping me with my writing. I would like to thank Paul Thompson for providing IT support and Victoria Tolton for providing administrative support in times of need.

I would like to thank my research group members Eishiro Higo, Sreehari Ramachandra Prabhu, Bo Li, Bin Liu, Min Xiang, Wanxin He, Nigel Fluegel, Arjun Gandhi, and Simarpreet Singh for their constant support and encouragement. I would like to thank Subhajit Majhi, Rakesh Ranjan, Bipin K. Gupta, Pampa Dey, Piyius Raj Singh, Avishek Paul, Wanis Nafo, Anurag Anand, and Nivas Ramachandiran for their endless support and encouragement. Special thanks are reserved for my friends Rithwik Kodamana, Harshwin Venugopal, Yamuna Jayan, and Sruthy Paul with whom I shared many unforgettable memories. A heartfelt thanks to Rithwik for his brotherly support through all these years.

I would like to thank my wife Sukanya for her love and constant support, and for keeping me sane over the past few years. To my parents and brother Kinshuk, thank

you for your love, support, and unwavering belief in me. Without you, I would not be the person I am today. I wish to thank my parents-in-law, sister-in-law Ankana, and cousin Saibal for their love and support.

Dedication

This thesis is dedicated to my parents.

Contents

List of Figures

List of Tables

List of Abbreviations

ABC	Approximate Bayesian computation
ABC-HMC	Approximate Bayesian computation using Hamiltonian Monte Carlo
ABC-MCMC	Approximate Bayesian computation using Markov chain Monte Carlo
ABC-RS	Approximate Bayesian computation rejection sampler
ABC-SS	Approximate Bayesian computation using Subset Simulation
ACF	Autocorrelation function
CANDU®	Canadian deuterium uranium
CC	Chemical cleaning
CDF	Cumulative distribution function
CI	Credible interval
COV	Coefficient of variation
DOF	Degree of freedom
EFPY	Effective full power years
FAC	Flow-accelerated corrosion
GR	Gelman-Rubin
HMC	Hamiltonian Monte Carlo

iid	Independent and identically distributed
LMER	Linear mixed-effects regression
L-MCMC	Likelihood-based Markov chain Monte Carlo
MCMC	Markov chain Monte Carlo
MH	Metropolis-Hastings
MHMC	Modified Hamiltonian Monte Carlo
MLE	Maximum likelihood estimation
MM	Modified Metropolis
NHPP	Non-homogeneous Poisson process
PDF	Probability density function
POD	Probability of detection
TWD	Through-wall depth
WL	Water lancing

List of Notations

$\lvert\mathbf{A}\rvert$	Determinant of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	Inverse of matrix $\mathbf{A}$
$\mathbf{A}^{\top}$	Transpose of matrix $\mathbf{A}$
$\mathcal{A}(\mathbf{x}, \mathbf{x}^*)$	Acceptance probability for transitioning from state $\mathbf{x}$ to $\mathbf{x}^*$
$\mathcal{B}_{ij}$	Bayes factor represented as $\frac{f(\mathcal{M}_i\|\mathbf{D}_{\text{obs}})/f(\mathcal{M}_j\|\mathbf{D}_{\text{obs}})}{f(\mathcal{M}_i)/f(\mathcal{M}_j)}$ for models $\mathcal{M}_i$ and $\mathcal{M}_j$
$\text{Cov}(X, Y)$	Covariance of two random variables X and Y
$\mathbb{E}[X]$	Expectation of random variable X
$f_X(x)$	PDF of random variable X
$f_X(x \mid \boldsymbol{\theta})$	PDF of random variable X conditioned on parameter vector $\boldsymbol{\theta}$
$f(\boldsymbol{\Theta})$	Prior distribution of the vector of model parameters $\boldsymbol{\Theta}$
$f(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$	Posterior distribution of parameter vector $\boldsymbol{\Theta}$ given observed data $\mathbf{D}_{\text{obs}}$
$f_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$	ABC posterior of $\boldsymbol{\Theta}$ given tolerance ϵ and observed data $\mathbf{D}_{\text{obs}}$
$f_{\max}(h; t)$	PDF of the maximum pit depth
$F_{\max}(h; t)$	CDF of the maximum pit depth
$\mathcal{G}(a, b)$	Gamma distribution with shape parameter a and scale parameter b
$\Gamma(s)$	Gamma function defined as $\Gamma(s) = \int_0^{+\infty} t^{s-1} e^{-t} dt$

$H(\boldsymbol{\Theta}, \mathbf{p})$	Hamiltonian of a dynamical system with position $\boldsymbol{\Theta}$ and momentum $\mathbf{p}$
$K(\mathbf{p})$	Kinetic energy of a dynamical system with momentum $\mathbf{p}$
$\mathcal{K}(\mathbf{x}^* \mid \mathbf{x})$	Transition kernel for transitioning from state $\mathbf{x}$ to $\mathbf{x}^*$
$\mathcal{L}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$	Likelihood function of parameter vector $\boldsymbol{\Theta}$ given observed data $\mathbf{D}_{\text{obs}}$
$\mathcal{L}_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$	Approximate ABC likelihood given tolerance ϵ and observed data $\mathbf{D}_{\text{obs}}$
$\mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta})$	Forward model for data simulation in ABC
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2
$p(h)$	POD function for pit depth h
$p_{ij}(m, n)$	Probability for transitioning from state $\mathbf{X}(t_n) = \mathbf{s}_j$ to state $\mathbf{X}(t_m) = \mathbf{s}_i$
$P(m, n)$	Transition probability matrix for transitioning from $\mathbf{X}(t_n)$ to $\mathbf{X}(t_m)$
$\mathbb{P}(A)$	Probability of an event A
$q(\mathbf{x}^* \mid \mathbf{x})$	Proposal density for proposing a move from state $\mathbf{x}$ to $\mathbf{x}^*$
Q_ϵ	ABC distance percentile criterion for tolerance selection
$\rho(\mathbf{D}, \mathbf{D}_{\text{obs}})$	Distance function between simulated data $\mathbf{D}$ and observed data $\mathbf{D}_{\text{obs}}$
$U(\boldsymbol{\Theta})$	Potential energy of a dynamical system with position $\boldsymbol{\Theta}$
$\mathcal{U}[a, b]$	Uniform distribution with lower bound a and upper bound b
$\text{Var}[X]$	Variance of random variable X
$X \sim f(x)$	Random variable X follows probability distribution $f(x)$
$\mathbb{1}[A]$	Indicator function: one, if condition A is satisfied, and zero, otherwise

Chapter 1

Introduction

1.1 Background

To support risk-based life cycle management of engineering systems, there is a need for periodic monitoring of the system and system components, data collection, data analysis, component-lifetime prediction, repair/refurbishment works, replacement planning, and finally, decommissioning of existing systems after their end of service life. The goal, however, is to avoid frequent failures of engineering components and maintain reliable operation during the service life of the system. In the light of risk-based life cycle management, this study seeks to find answers to the challenging problems that arise in the degradation model selection, calibration, and prediction – a subset of the bigger problem that involves the implementation and execution of such management strategies for large engineering systems.

Structural components are often subjected to various types of degradation processes such as corrosion, crack, fatigue, and creep, depending on the condition of the surrounding environment. For example, the Canadian deuterium uranium (CANDU®) reactors (see Figure 1.1) contain numerous small diameter pipes called feeder pipes that

carry heavy water coolant to the reactor core. The flow-accelerated corrosion (FAC) is a major form of degradation seen in these pipes [60, 77]. These pipes are a part of the primary heat transport system of the nuclear power plant, which also contains steam generators made up of several thousand thin-walled tubes. These tubes help in producing steam by transferring the heat carried by the heated coolant in feeder pipes. Similar to the feeder pipes, the steam generator tubes mostly suffer from pitting corrosion [148]. As another example, the reactor core has a large number of fuel channels, also known as the pressure tubes, carrying the nuclear fuel, that suffer mostly from the irradiation creep [105]. Although the basic mechanics of these degradation mechanisms is well studied in the literature, high variability of these degradation mechanisms is seen in practice due to a combination of factors such as the material and geometric shape of a structural component and its surrounding environment.

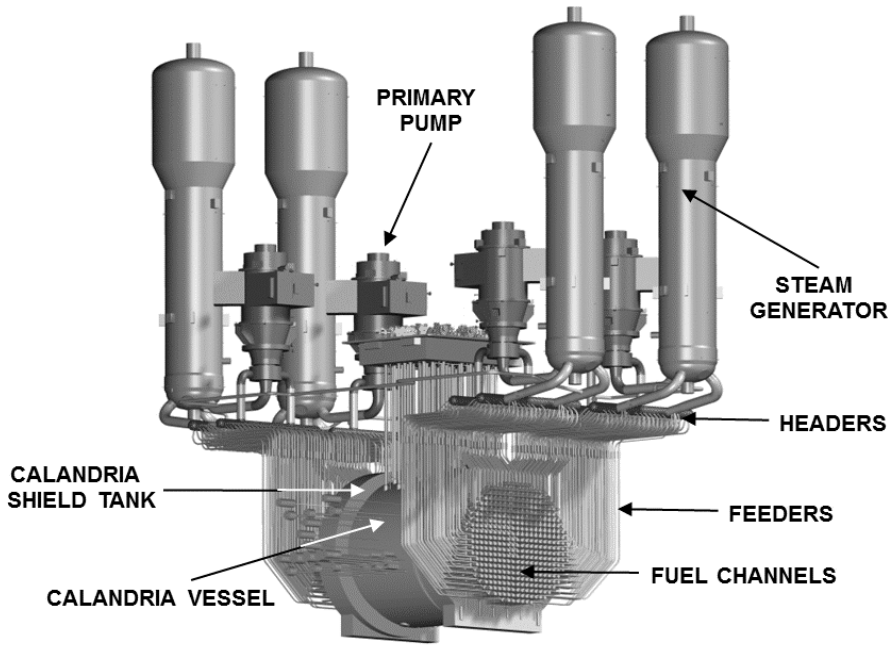


Figure 1.1: Layout of a typical CANDU® reactor and its primary heat transport system [47]. (Image reproduced with permission.)

Periodic in-service inspections are carried out in nuclear power plants to measure the extent of degradation in a component. Various non-destructive inspection tools such as ultrasonic and eddy current probes are generally used for these inspections. The inspection data are then used to identify the heavily degraded components, plan component replacement, and set next inspection schedules. Due to variable operating conditions, different components experience different rates of degradation. This variability related to the degradation in the component population can be modeled using stochastic degradation models.

Given a degradation process $X(t)$, an example of a cumulative model for degradation can be written as $X(t_{n+1}) = X(t_n) + \Delta X$, where ΔX is the degradation increment within the time interval (t_n, t_{n+1}). The degradation process can be appropriately modeled using a stochastic process (e.g., gamma process). Sample paths of a typical degradation growth process are shown in Figure 1.2. Let us assume that each sample path belongs to a specific component. It can be observed that a few sample paths have

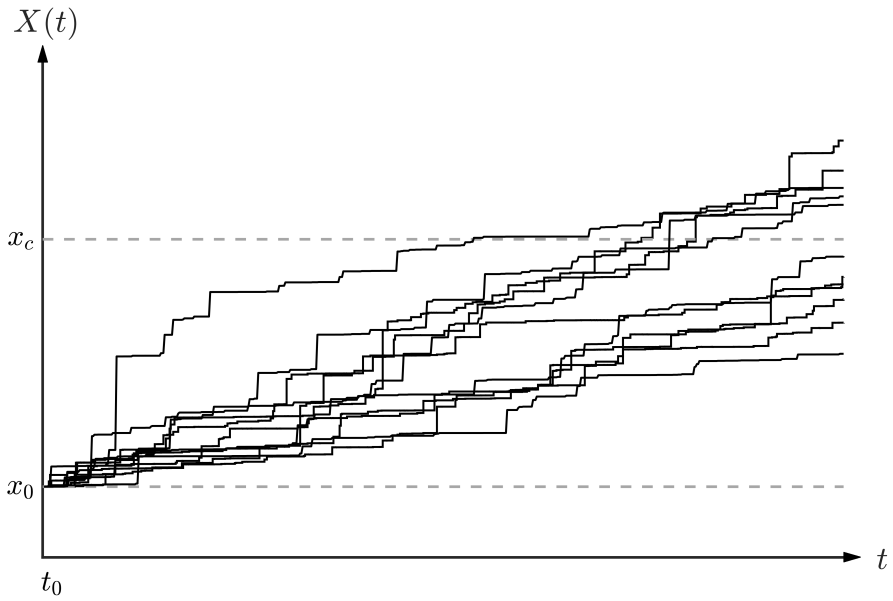


Figure 1.2: Sample paths of a typical degradation growth process.

crossed the threshold x_c which indicates that the specific components have turned substandard. On the other hand, those sample paths which have not crossed x_c within the given time frame indicate that the specific components can be continued to be used until they become substandard. This is one type of degradation process; other types include degradation/flaw generation (e.g., pitting corrosion [148]) and two-phase degradation growth, i.e., a degradation process that involves a change point [106]).

In stochastic degradation modeling, the main goal is to estimate the parameters of degradation models from inspection data. However, the estimation task often becomes challenging due to the effect of inspection and sampling uncertainties on the model parameters. Inspection uncertainties are introduced by imperfect inspection data, such as imperfect flaw size measurements or non-detection of small defects in components. On the other hand, limited inspection data is the main reason for sampling uncertainty, which is caused due to inaccessibility of nuclear systems for high levels of radiation and large costs associated with remote data collection methods.

1.2 Motivation

The inspection data could be influenced by two main inspection uncertainties. Firstly, the electronic inspection tools, by their very nature, do not measure the actual or true size of any defect in a component, but instead, give only imperfect or noisy measurements. Moreover, these electronic tools fail to detect small defects in a component under a certain detection-threshold. Sometimes, these imperfect or noisy measurements become so significant that they make the whole process of degradation assessment, modeling, and prediction very challenging and uncertain. Thus, the inspection related uncertainties due to imperfect measurements or non-detection of small defects can not be ignored. Secondly, nuclear facilities are generally inaccessible due to the

presence of high levels of radiation. Thus, remote data collection methods are difficult to employ and quite expensive. For these two reasons, often only a few components are inspected and only a few component-wise measurements are generally taken. The limited inspection data eventually introduces sampling uncertainty to the parameters of the degradation model, which adds up to inspection uncertainty and makes the degradation assessment more complicated and ambiguous.

There is certainly a need for a proper approach to assess the limited noisy degradation data and estimate the parameter uncertainties in stochastic degradation modeling. The two most popular methods for parameter estimation are the maximum likelihood estimation (MLE) method and the Bayesian inference method. MLE is based on the frequentist approach to inference, which treats the unknown parameters of a model as fixed quantities or constants. Frequentist inference is related to the frequentist interpretation of probability, according to which probabilities are presumed as limiting frequencies of outcomes after infinite hypothetical repetitions of an experiment generating statistically independent results [140]. In this approach, the specific estimators are assessed under repeated sampling of the available data, and the parameter uncertainties are represented in terms of confidence intervals obtained from the numerical estimates of the unknown model parameters. The Bayesian approach, on the other hand, interprets probabilities as subjective, i.e., based on an individual's personal judgment/experience, and dependent on the available data/information. Thus, the Bayesian probabilities of a specific event can vary among individuals [140]. In this inference approach, all unknown model parameters are assumed as random variables, and the associated (posterior) probability distributions represent the uncertainties of these parameters.

Although MLE is well suited for large amounts of available data, the Bayesian approach can handle uncertainties more efficiently with only a small amount of available

data [20]. The main reason behind this anomaly is that the likelihood function formed using limited data is often unable to generate distinct peaks on its surface, leading to problems related to the convergence of the maximum likelihood estimates. As a result, the maximum likelihood estimates of the parameters show larger uncertainty in terms of large confidence intervals. Besides, in Bayesian inference, prior beliefs (represented by a prior distribution) about the model parameters are formally updated using the Bayes' theorem. The updated posterior belief or posterior distribution accurately represents the parameter uncertainty. The posterior distribution, in words, can be written as

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Normalizing constant}} \tag{1.1}$$

where the likelihood function represents the probability density of the observed data that depends on the underlying model and the normalizing constant represents an integration over the entire range of the parameter space. Thus, the Bayesian inference method emerges as a powerful tool for handling large uncertainties in the inspection data.

A disadvantage of the Bayesian inference method is that it is difficult to use in engineering practice due to its computational complexities. Although the computational difficulty of computing the normalizing constant can be avoided by implementing the Markov chain Monte Carlo (MCMC) methods, the evaluation of the likelihood function numerous times still remains as a computational burden for the Bayesian inference approach. For a detailed description on the MCMC methods, the reader is referred to Chapter 3. The primary reason behind this computational issue is that the sample likelihoods of stochastic degradation models often involve large numbers of convolution integrals or high-dimensional infinite sums or, sometimes, a combination of both. For instance, see Chapter 2, where the likelihood functions of standard degradation mod-

els are derived for noisy data. These types of complex likelihood functions impose a severe computational burden on the traditional Bayesian computation schemes. The computational complexity of the Bayesian inference method often compels one to discard imperfect data and only perform the analysis based on healthy or less noisy data. An analyst may choose to use simple models so that they can plug-in conjugate priors and obtain posterior distributions analytically without any further effort. But these simple models may not represent the degradation mechanism well.

After analyzing the computational issues related to the standard Bayesian inference method, it can be stated that we need a likelihood-free treatment for the parameter estimation problem in degradation modeling. To overcome the drawbacks of computing a likelihood function, and expand our domain of model selection, the novel approximate Bayesian computation (ABC) technique [82, 124] can be implemented. ABC is a likelihood-free Bayesian computation algorithm that completely avoids likelihood computation by using the idea of forward simulation. The basic concepts of the ABC method are presented in Chapter 4. If data simulation from a forward model is computationally cheap, then ABC turns out to be an efficient alternative to the traditional Bayesian inference method. Among the many advantages of ABC, this method is intuitive, simple-to-understand, and easy-to-implement – making it a perfect choice for the Bayesian inference of stochastic degradation models, particularly for practical applications.

1.3 Research Gaps

Literature on stochastic degradation modeling is vast. However, if one concentrates on a subset of this literature that includes degradation modeling using Bayesian methods, the following research gaps can be noticed:

1. Compared to the MLE approach, the literature on the Bayesian approach is limited due to the numerical complexities involved in the implementation of this method. Thus, a simple and easy-to-implement Bayesian computational method is needed, which can handle the parameter inference problem efficiently in the stochastic degradation modeling process.

2. An important problem in the degradation modeling procedure is the model selection process. Nguyen et al. [96] proposed a sound methodology for degradation model selection using the MLE method. However, they did not consider measurement noise in their analysis. The MLE approach can be a better option if the data is noise-free and of large volume. But, for a small amount of data, the Bayesian model selection approach is the most suitable approach. The Bayesian model selection method has a greater advantage over other approaches because this approach considers the parameter uncertainties more naturally through posterior distributions and automatically accounts for the number of parameters in a model (i.e., penalize a model if it has more parameters). There is again a need for an efficient and easy-to-implement method that can avoid the complexity of the traditional Bayesian model selection procedure.

3. There is a gap in the literature on how to integrate the uncertainties in degradation model parameters for population lifetime prediction and estimation of survival probability or reliability of a system of components.

The ABC method is a simulation approach for estimating parameters of a model. ABC compares simulated data sets with the observed data set using a distance function, and obtains the posterior parameter samples by accepting or rejecting the corresponding parameters based on a tolerance threshold on the distance values. To tailor the ABC algorithm according to the needs of stochastic degradation modeling, the

following investigations are necessary:

1. One major issue in the ABC method is that there is a lack of practical guideline for the selection of the distance function. Moreover, different kinds of degradation mechanism comes with different kinds of inspection data. For example, flaw growth data contain multiple measurements of flaw growths from different components, whereas flaw generation data will contain the number of flaws and their measured depths. Thus, it is necessary to identify the special features of the degradation data at hand, and develop distance functions accordingly to produce the best results from ABC in stochastic degradation modeling.

2. Degradation data are often imperfect and limited in amount. In such a situation, one needs a method that properly takes the measurement errors and detection issues in data into account. Except for the ABC method proposed by Wilkinson [145], none of the existing ABC algorithms properly address the issue of measurement errors and detection issues. However, Wilkinson's ABC needs a well-defined error distribution that may not be available when dealing with a practical data set. Thus, the existing ABC algorithms need to be modified to account for imperfect data which can be done at the model simulation stage.

3. One needs to select a tolerance threshold for the ABC to work. The tolerance threshold determines whether the simulated data is close enough to the observed data or not. A smaller tolerance threshold gives better accuracy, but the question remains how small is small enough? Hence, there is a need to investigate on the selection of the tolerance threshold for modeling degradation data using ABC.

4. The ABC method has a high rejection rate when a smaller tolerance threshold is chosen. Thus, various sampling algorithms, such as MCMC and sequential

Monte Carlo, can be used within the ABC framework to reduce the rejection rate. However, these sampling schemes either have high repetitions of samples (e.g., ABC-MCMC) or high number of simulation levels (e.g., sequential ABC) when many parameters are involved in the process. In fact, the sequential ABC method may get stuck in a particular level for a very long period – making the parameter estimation process sluggish. These issues need to be investigated particularly for stochastic degradation modeling since it involves a varying number of parameters depending on the underlying model.

1.4 Research Objectives

The principle objective of this thesis is two-fold:

1. To develop a unified framework for likelihood-free inference of parameters of stochastic degradation models.

2. To integrate parameter uncertainties into model prediction, lifetime estimation, and survival probability calculation of components.

To achieve this goal, the research objectives of this study are divided into many sub-steps as follows:

- **Algorithmic improvements:**

    1. To identify potential candidates for the ABC distance function and investigate their efficacy with various kinds of degradation data.

    2. To develop an ABC framework that effectively deals with measurement noise and detection errors in degradation data.

3. To develop an advanced ABC scheme for degradation modeling that reduces the rejection rate in the basic ABC scheme, uses less model simulations, and provides a proper guideline for the selection of the tolerance threshold.

- **Application:**

1. To explore and compare the computational difficulties posed by the implementation of both likelihood-based and likelihood-free Bayesian estimation schemes for standard stochastic degradation models (e.g., random rate, regression, gamma process, and Poisson process).
2. To develop and implement a Bayesian model selection framework for degradation data using the likelihood-free approach.
3. To devise strategies for integrating parameter uncertainties into model prediction, lifetime estimation, and estimation of survival probabilities for the system and individual components.

1.5 Organization of the Thesis

The thesis is organized as follows:

- Chapter 1 presents the research goals of this study along with the background, the motivation, and the research gaps present in the literature of degradation modeling.
- Chapter 2 presents the basic properties of the standard degradation models under consideration and the corresponding model likelihoods derived for degradation data that are impacted by various types of inspection uncertainties.

- Chapter 3 presents the popular MCMC methods and shows various MCMC algorithms for Bayesian computation.

- Chapter 4 presents the basic idea of likelihood-free inference along with various standard ABC algorithms. A number of algorithms are developed and the details of their implementation are also presented in this chapter.

- Chapter 5 presents various case studies on corrosion growth modeling as practical applications of the proposed ABC framework using degradation data from CANDU® nuclear power plants.

- Chapter 6 presents a case study on flaw generation modeling using pitting corrosion data from the CANDU® steam generator tubes.

- Chapter 7, finally, presents the conclusions of the study and discusses the directions for future research.

Chapter 2

Degradation Models

2.1 Introduction

Stochastic process models are useful for modeling degradation processes due to their ability to imitate flaw generation and growth directly by using a collection of random variables, where the corresponding distributions are functions of time. An example of early application of stochastic process models in degradation can be found in reference [17], where the authors have used the renewal process model to study the fatigue damage of structures under dynamic loads. The popular Paris-Erdogan law of fatigue crack growth is in fact a non-linear general path model [102]. Modeling bridge deck deterioration [80, 81], rock rubble replacement [139], and water pipe degradation [93] are a few examples of current applications of the stochastic process models. Moreover, for modeling the degradation of nuclear power plant components, such as FAC in feeder pipes [28, 59–61, 100, 147], fretting wear and pitting flaws in steam generator tubes [26, 148], stochastic process models are widely accepted.

This study considers four popular probabilistic/stochastic degradation models: the random variable model, the gamma process model, the linear mixed-effects regression

(LMER) model, and the non-homogeneous Poisson process (NHPP) model. The first three models, random variable, gamma process, and LMER, are used to model the flaw growth phenomena (e.g., FAC-induced wall thickness loss of piping components), whereas the NHPP model is used to characterize the flaw generation process in components over time (e.g., pitting corrosion in steam generator tubes).

2.2 Flaw Growth Models

2.2.1 Random Variable Model

The basic idea of the random variable model is to capture the variability related to the degradation growth rates of different components in a component population. This model is generally applied to problems such as modeling corrosion and wear phenomena [45, 65, 78, 98], where the temporal uncertainty related to the degradation process is not significant [99].

Basic Properties

The random variable model, also known as the general path model, is defined by a deterministic function that has random parameters. Suppose $X(t)$ is the degradation state of a structural component at time t. According to this model, $X(t)$ can be represented as

$$X(t) = g(t; \boldsymbol{\Theta}) \tag{2.1}$$

where $g(t; \boldsymbol{\Theta})$ is a function of time t and $\boldsymbol{\Theta}$ is a vector of random variables. The flaw growths of individual components are represented by $g(t; \boldsymbol{\theta}_k)$, where $\boldsymbol{\theta}_k$ is a realization of $\boldsymbol{\Theta}$. The most basic version of the random variable model is the linear random rate model. Assuming $X(0) = 0$, the random rate model can be described by the equation

$X(t) = Rt$, where the random variable $R > 0$ is used to model the degradation growth rate in a system of components. Several flaw growth paths simulated from the random rate model are shown in Figure 2.1. Once the distribution of the degradation rate R is

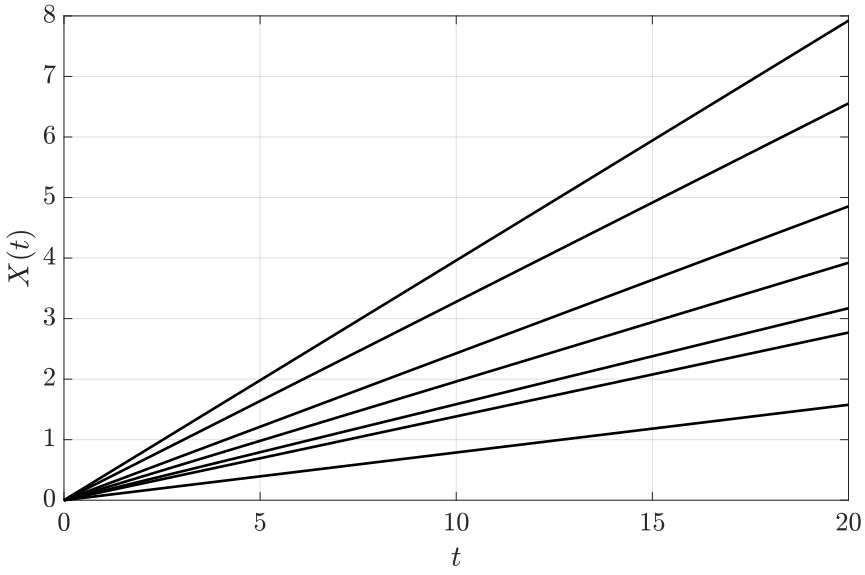


Figure 2.1: Simulated flaw growth paths of the random rate model.

known, one can determine the distribution of the degradation process at time t. The mean and variance of $X(t)$ can be computed as

$$\mathbb{E}[X(t)] = t\mathbb{E}[R], \ \mathrm{Var}[X(t)] = t^2\mathrm{Var}[R] \tag{2.2}$$

where $\mathbb{E}[\bullet]$ and $\mathrm{Var}[\bullet]$ are the expectation and variance operators, respectively. Sometimes, due to manufacturing (e.g., bending) or welding operations, initial conditions of structural components (e.g., initial wall thicknesses of pipes) are not known precisely. This leads to the inclusion of an additional random variable A to the model that represents the initial condition of a structural component. Thus, assuming $X(0) = A$, the

random rate model can be written as

$$X(t) = A + Rt \tag{2.3}$$

where A is assumed to follow a distribution whose parameters are unknown and needs to be estimated.

Likelihood Function

Suppose inspection data from N number of components are available. The ith component is inspected at times, $t_{i1}, t_{i2}, \cdots, t_{im_i}$, where m_i is the total number of inspections conducted on the same ith component. Let us assume that the true degradation measurements are represented as $x_{i1}, x_{i2}, \cdots, x_{im_i}$ and the observed degradation measurements as $y_{i1}, y_{i2}, \cdots, y_{im_i}$. The observed degradation measurements are often masked with measurement noise added by the inspection tools. For instance, ultrasonic probes that are used for pipe inspections add random sizing error to the inspection data. Consequently, using the basic model for degradation growth

$$Y(t) = X(t) + Z = A + Rt + Z \tag{2.4}$$

the measurement model for the ith component at the jth measurement time can be written as

$$y_{ij} = a_i + r_i t_{ij} + z_{ij}; \quad a_i \sim f_A(\bullet \mid \boldsymbol{\theta}_1),\ r_i \sim f_R(\bullet \mid \boldsymbol{\theta}_2),\ z_{ij} \overset{\text{iid}}{\sim} f_Z(\bullet) \tag{2.5}$$

where a_i is a realization of the random variable A with distribution $f_A(\bullet \mid \boldsymbol{\theta}_1)$, $r_i > 0$ is a realization of the degradation growth rate R with distribution $f_R(\bullet \mid \boldsymbol{\theta}_2)$, and finally, the measurement errors z_{ij} are independent and identically distributed (iid) random

variables with the distribution $f_Z(\bullet)$. Both parameters A and R are independent to each other and their PDFs are conditioned on the vector of distribution parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, respectively.

All measurements of an ith flaw can be represented by a vector $\mathbf{y}_i = \{y_{i1}, y_{i2}, \cdots, y_{im_i}\}^\top$, where $\top$ denotes transposition. The initial degradation a_i and the corrosion rate r_i are constant for the i^{th} component. For given a_i and r_i, the true degradation growth $x_{ij} = a_i + r_i t_{ij}$ is a constant. Thus, the distribution of the measurements y_{ij}, conditioned on a_i and r_i, is solely dependent on the distribution of the measurement noise $f_Z(\bullet)$, and can be written as,

$$f_{Y_{ij}}(y_{ij} \mid a_i, r_i) = f_Z(y_{ij} - a_i - r_i) \tag{2.6}$$

Assuming independency between the degradation measurements, the joint density of the measurements of i^{th} component, conditioned on a_i and r_i, can written as

$$f_{\mathbf{Y}_i}(\mathbf{y}_i \mid a_i, r_i) = \prod_{j=1}^{m_i} f_{Y_{ij}}(y_{ij} \mid a_i, r_i) = \prod_{j=1}^{m_i} f_Z(y_{ij} - a_i - r_i) \tag{2.7}$$

Thus, the sample likelihood for the i^{th} component's data can be written as

$$\begin{aligned} \ell_i(\boldsymbol{\Theta} \mid \mathbf{y}_i) &= \int_{a_i}\int_{r_i} f_{\mathbf{Y}_i}(\mathbf{y}_i \mid a_i, r_i) f_A(a_i \mid \boldsymbol{\theta}_1) f_R(r_i \mid \boldsymbol{\theta}_2) da_i dr_i \\ &= \int_{a_i}\int_{r_i} \Big\{ \prod_{j=1}^{m_i} f_Z(y_{ij} - a_i - r_i) \Big\} f_A(a_i \mid \boldsymbol{\theta}_1) f_R(r_i \mid \boldsymbol{\theta}_2) da_i dr_i \end{aligned} \tag{2.8}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1^\top, \boldsymbol{\theta}_2^\top\}^\top$. Assuming the component-wise measurements to be independent, the sample likelihood using data from all components can be written as

$$\mathcal{L}(\boldsymbol{\Theta} \mid \mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N) = \prod_{i=1}^{N} \ell_i(\boldsymbol{\Theta} \mid \mathbf{y}_i)$$
$$= \prod_{i=1}^{N} \int_{a_i} \int_{r_i} \left\{ \prod_{j=1}^{m_i} f_Z(y_{ij} - a_i - r_i) \right\} f_A(a_i \mid \boldsymbol{\theta}_1) f_R(r_i \mid \boldsymbol{\theta}_2) da_i dr_i \tag{2.9}$$

As it can be observed, the likelihood function of the random rate model is a product of N two-dimensional integrals. For parameter estimation, this likelihood function needs to be evaluated several times. The numerical integration can be performed using Monte Carlo simulation methods [76].

2.2.2 Gamma Process Model

The stochastic gamma process is used to model a degradation process that shows temporal uncertainties to a level which is significant for model prediction. During the last four decades, the stochastic gamma process has been extensively used to model various degradation processes in engineering structures and components. In a very interesting paper by Abdel Hameed [1], gamma process was first proposed as a proper model for modeling stochastic degradation. Examples of using gamma process in various degradation processes such as corrosion, concrete creep, crack growth, fatigue, and chloride attack in concrete structures can be found in references [8, 35, 46, 71, 74].

Basic Properties

The stochastic gamma process model can be used to characterize a monotonically non-decreasing degradation process. The key assumption of the gamma process model is that a degradation process $X(t)$ is the result of accumulation of several small and independent random degradation growths. To be more specific, the degradation process $X(t)$ will have the following properties [138]:

1. At time $t = 0$, $X(t) = 0$.

2. For $0 \leq t_1 \leq t_2 \leq \cdots \leq t_n$, all increments of $X(t)$, i.e., $X(t_1) - X(0), X(t_2) - X(t_1), \cdots, X(t_n) - X(t_{n-1})$ are independent random variables.

3. An increment $\Delta X = X(t + \Delta t) - X(t)$, for $\Delta t \geq 0$, follows gamma distribution with the following probability density function (PDF):

$$\begin{aligned} f_{\Delta X}(\Delta x) &= \frac{(\Delta x/\beta)^{a(t+\Delta t)-a(t)-1}}{\beta\,\Gamma(a(t+\Delta t)-a(t))}\exp(-\Delta x/\beta);\;\; \Delta x \geq 0 \\ &= \mathcal{G}(a(t+\Delta t)-a(t),\beta) \end{aligned} \tag{2.10}$$

   where $\mathcal{G}(\bullet,\bullet)$ is the gamma PDF, $a(t) > 0$ is the shape parameter, $\beta > 0$ is the scale parameter, and $\Gamma(s) = \int_0^{+\infty} t^{s-1}e^{-t}dt$ is the complete gamma function.

The shape parameter $a(t)$, for $t \geq 0$, is a non-decreasing, right-continuous, real valued function, and at $t = 0$; $a(t) = 0$. $a(t)$ is assumed to be proportional to a power law, $a(t) \propto t^\eta \Rightarrow a(t) = \alpha t^\eta$, where $\alpha > 0$ and $\eta > 0$ are constants [138]. A gamma process model with the parameter $\eta = 1$ is called stationary, whereas a gamma process having $\eta \neq 1$ is called a non-stationary process. Figure 2.2 shows several simulated flaw growth paths from stationary and non-stationary gamma process models with $\alpha = 1.5$ and $\beta = 0.2$. The gamma distribution has a special feature that makes the sum of two gamma random variables also a gamma random variable. Following the same argument, the cumulative degradation $X(t)$ at time t can be considered a gamma distributed random variable with PDF $\mathcal{G}(\alpha t^\eta, \beta)$ [79]. To calculate the mean and variance of the process $X(t)$, the following expressions can be directly used:

$$\mathbb{E}[X(t)] = \alpha t^\eta \beta,\;\; \text{Var}[X(t)] = \alpha t^\eta \beta^2 \tag{2.11}$$

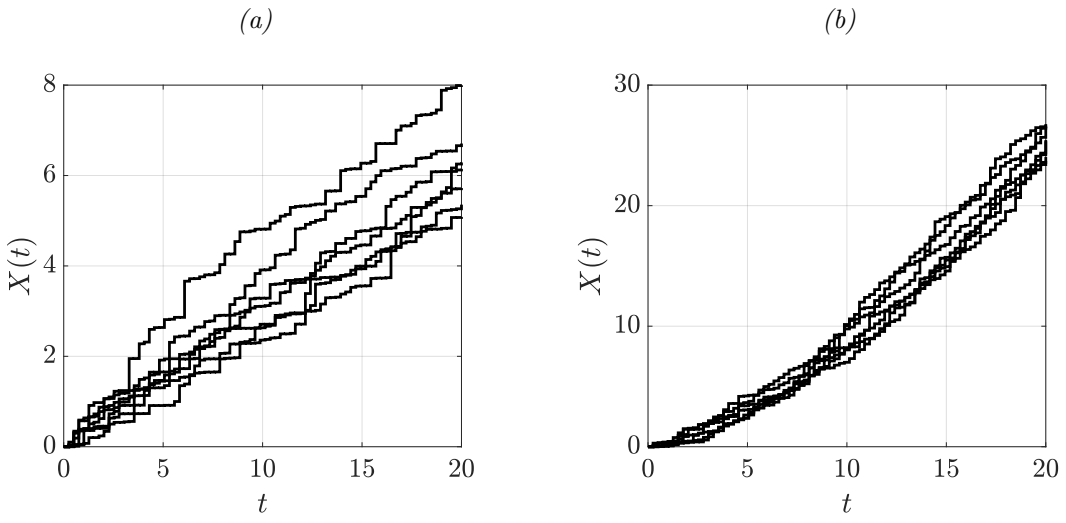


Figure 2.2: Simulated flaw growth paths of the gamma process model for (a) $\eta = 1$ and (b) $\eta = 1.5$.

Likelihood Function

Under the assumption of an additive noise model, the measured degradation $Y(t)$ is represented as a sum of random initial degradation A, true degradation $X(t)$, and measurement noise Z as [78]:

$$Y(t) = A + X(t) + Z \tag{2.12}$$

Thus, a noisy degradation measurement of component i at time j can be represented as $y_{ij} = a_i + x_{ij} + z_{ij}$, where a_i ($y_{i0} = a_i$) is the initial degradation, x_{ij} is the true degradation growth at time t_{ij}, and z_{ij} ($z_{i0} = 0$) is the random sizing error. The unknown initial degradation a_i (constant for each component but variable over the entire population) is a realization of A with distribution $f_A(\bullet)$, the true degradation x_{ij} is a realization of the gamma degradation process $X(t)$, and the sizing error z_{ij} is a realization of the iid random variable Z with distribution $f_Z(\bullet)$.

Suppose degradation monitoring data are collected from repeated inspections of a group of N components at various time intervals. For an ith component, true values of degradation at different inspection times $t_{i0}, t_{i1}, \cdots, t_{im_i}$ are denoted as $x_{i0}, x_{i1}, \cdots, x_{im_i}$, where $x_{i0} = 0$ at time $t_{i0} = 0$ and m_i is the total number of inspections. The degradation growth over a time interval $\Delta t_{ij} = t_{ij} - t_{i,j-1}$ is denoted as $\Delta x_{ij} = x_{ij} - x_{i,j-1}$, where $j = 1, 2, \cdots, m_i$. The increment $\Delta x_{ij} = x_{ij} - x_{i,j-1}$ is a realization of the random variable ΔX_{ij} which follows the gamma distribution $\mathcal{G}(\alpha \Delta t_{ij}^{(\eta)}, \beta)$, where $\Delta t_{ij}^{(\eta)} = t_{ij}^{\eta} - t_{i,j-1}^{\eta}$. All degradation increments for an ith component are denoted as a vector $\Delta \mathbf{x}_i = \{\Delta x_{i1}, \Delta x_{i2}, \cdots, \Delta x_{im_i}\}^{\top}$ with the probability distribution:

$$\begin{aligned} f_{\Delta \mathbf{X}_i}(\Delta \mathbf{x}_i \mid \boldsymbol{\Theta}) &= \prod_{j=1}^{m_i} \mathcal{G}(\alpha \Delta t_{ij}^{(\eta)}, \beta) \\ &= \prod_{j=1}^{m_i} \frac{(\Delta x_{ij}/\beta)^{\alpha(t_{ij}^{\eta} - t_{i,j-1}^{\eta})-1}}{\beta\, \Gamma(\alpha(t_{ij}^{\eta} - t_{i,j-1}^{\eta}))} \exp(-\Delta x_{ij}/\beta) \end{aligned} \tag{2.13}$$

where $\boldsymbol{\Theta} = \{\alpha, \eta, \beta\}^{\top}$ is the vector of unknown model parameters.

Assuming y_{ij} to be a realization of the random variable Y_{ij}, the sample likelihood for degradation measurement data collected from an ith component can be written in terms of the joint distribution of $Y_{i1}, Y_{i2}, \cdots, Y_{im_i}$, which are dependent variables. Thus, it is more convenient to write the sample likelihood in terms of the joint distribution of degradation increments [77, 146]. Let us denote a measured degradation increment for an ith component over a time interval Δt_{ij} as $\Delta y_{ij} = y_{ij} - y_{i,j-1}$ and the corresponding increment in the noise as $\Delta z_{ij} = z_{ij} - z_{i,j-1}$. Measured values of incremental degradation and corresponding increment in noise are denoted in vector forms as $\Delta \mathbf{y}_i = \{\Delta y_{i1}, \Delta y_{i2}, \cdots, \Delta y_{im_i}\}^{\top}$ and $\Delta \mathbf{z}_i = \{\Delta z_{i1}, \Delta z_{i2}, \cdots, \Delta z_{im_i}\}^{\top}$. Note that $\Delta y_{i1} = y_{i1} - y_{i0}$ is unknown because $y_{i0} = a_i$ is a latent variable. The quantities $\Delta \mathbf{y}_i$ and $\Delta \mathbf{z}_i$ are assumed to be realizations of the random vectors $\Delta \mathbf{Y}_i$ and $\Delta \mathbf{Z}_i$,

respectively. The joint PDF of $\Delta\mathbf{Y}_i$ can be written through a convolution integral as

$$f_{\Delta\mathbf{Y}_i}(\Delta\mathbf{y}_i) = \int_{a_i}\int_{\Delta\mathbf{z}_i} f_{\Delta\mathbf{X}_i}(\Delta\mathbf{y}_i - \Delta\mathbf{z}_i \mid a_i) f_A(a_i) f_{\Delta\mathbf{Z}_i}(\Delta\mathbf{z}_i) da_i d(\Delta\mathbf{z}_i) \tag{2.14}$$

This convolution integral consists of the joint PDFs $f_{\Delta\mathbf{X}_i}(\Delta\mathbf{x}_i)$ and $f_{\Delta\mathbf{Z}_i}(\Delta\mathbf{z}_i)$, where $f_{\Delta\mathbf{Z}_i}(\Delta\mathbf{z}_i)$ is the joint PDF of the random vector $\Delta\mathbf{Z}_i$. Generally, the sizing error is assumed to follow a normal distribution with zero mean and σ_Z standard deviation, i.e., $f_Z(\bullet) = \mathcal{N}(0, \sigma_Z^2)$, where $\mathcal{N}(\bullet, \bullet)$ represents a normal density function. Thus, the joint PDF of the random vector $\Delta\mathbf{Z}_i$ follows a multivariate normal distribution given as (see Section A.1 for the derivation)

$$f_{\Delta\mathbf{Z}_i}(\Delta\mathbf{z}_i) = \frac{1}{(2\pi)^{m_i/2}|\mathbf{\Sigma}_{\Delta\mathbf{Z}_i}|} \exp\left\{ -\frac{1}{2}\Delta\mathbf{z}_i \mathbf{\Sigma}_{\Delta\mathbf{Z}_i}^{-1} \Delta\mathbf{z}_i^T \right\} \tag{2.15}$$

where $|\bullet|$ represents the determinant operator and $\mathbf{\Sigma}_{\Delta\mathbf{Z}_i}$ represents the variance-covariance matrix. Assuming that the degradation process is independent across the component population, the sample likelihood function for measured degradation data collected from all N components can be written as

$$\begin{aligned} \mathcal{L}(\mathbf{\Theta} \mid \Delta\mathbf{y}_1, \Delta\mathbf{y}_2, \cdots, \Delta\mathbf{y}_N) &= \prod_{i=1}^{N} f_{\Delta\mathbf{Y}_i}(\Delta\mathbf{y}_i) \\ &= \prod_{i=1}^{N} \int_{a_i}\int_{\Delta\mathbf{z}_i} f_{\Delta\mathbf{X}_i}(\Delta\mathbf{y}_i - \Delta\mathbf{z}_i \mid a_i) f_A(a_i) f_{\Delta\mathbf{Z}_i}(\Delta\mathbf{z}_i) da_i d(\Delta\mathbf{z}_i) \end{aligned} \tag{2.16}$$

It can be observed that the sample likelihood of the gamma process model is a product of N multi-dimensional integrals, which implies that it may be quite difficult to evaluate the likelihood function numerically.

2.2.3 Linear Mixed-Effects Regression Model

The random variable model, being a basic model, is only useful to estimate the characteristics of a system of components. To obtain degradation characteristics of individual components, one needs to analyze data from individual components which induces high uncertainty due to small volume of component-specific repeated measurements. Alternatively, the LMER model resolves this by systematically processing the inspection data by considering both the system-level fixed effects as well as component-level random effects. The model is well suited for pooling unbalanced data (i.e. different number of repeated measurements from components) from component-specific measurements across the component population to obtain robust estimates of the model parameters [94]. However, the LMER model has too many unknown parameters which make its calibration process challenging in case the data in hand is noisy.

Basic Properties

According to the LMER model [140], any degradation measurement y_{ij} can be represented as

$$y_{ij} = \beta_0 + \beta_1 t_{ij} + b_{0i} + b_{1i} t_{ij} + z_{ij} \tag{2.17}$$

where the true degradation growth x_{ij} is given by $(\beta_0+\beta_1 t_{ij}+b_{0i}+b_{1i}t_{ij})$. In the context of degradation modeling, the fixed effects parameters β_0 and β_1 represent initial degradation and degradation growth rate of the system of components, respectively; and the random effects parameters b_{0i} and b_{1i} represent the variation (from the fixed effects parameters) of initial degradation and degradation growth rate of the ith component, respectively. While the simple linear regression model allows only the fixed effects parameters, an extension of it, the LMER model, allows both fixed and random effects parameters to represent hierarchical data, meaning that the data contain information

at multiple levels (e.g., system-level, component-level). The fixed effects parameters represent the properties of a system as a whole, thus called system-level parameters. Whereas, the random effects parameters, when combined with the fixed effects parameters, represent the properties of individual components, thus called component-level parameters. Figure 2.3 shows simulated sample paths from the LMER model.

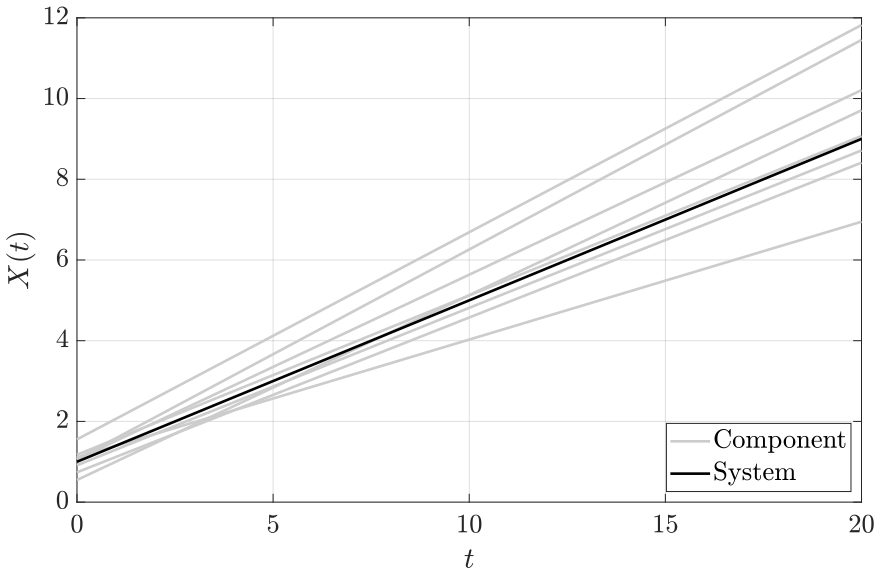


Figure 2.3: Simulated flaw growth paths of the LMER model.

The key assumptions of the LMER degradation model are as follows. The regression coefficients β_0 and β_1 are unknown constants. Through these two coefficients, the LMER model given by Equation 2.17 assumes that the degradation of a particular component is dependent not only on the component-specific parameters but also on the system-level parameters. This implies that, unlike a simple linear regression model, the resultant component-specific parameters are influenced by other component-specific repeated measurements as well as by the number of components inspected over time.

On the other hand, (b_{0i}, b_{1i}) are assumed to be iid bivariate normal random variables, i.e., $(b_{0i}, b_{1i}) \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_b)$, where $\mathbf{0}$ is the zero mean vector and $\mathbf{\Sigma}_b$ is the unknown

covariance matrix. Assuming the standard deviations of b_{0i} and b_{1i} to be σ_0 and σ_1, respectively, and the correlation coefficient to be ρ, the covariance matrix can be written as

$$\boldsymbol{\Sigma}_b = \begin{bmatrix} \sigma_0^2 & \rho\sigma_0\sigma_1 \\ \rho\sigma_0\sigma_1 & \sigma_1^2 \end{bmatrix} \tag{2.18}$$

The random effects parameters (b_{0i}, b_{1i}) and the noise term z_{ij} are independently distributed, which implies that the measurement error is independent of the true degradation.

Degradation data generally contain multiple repeated measurements from each component. Thus, any two observations from the same component are correlated, whereas the correlation is assumed to be zero for different components. These assumptions are automatically satisfied by the LMER model which gives the following covariance (Cov) structure for any two observations:

$$\mathrm{Cov}(y_{ij}, y_{hk}) = \begin{cases} \sigma_0^2 + \rho\sigma_0\sigma_1(t_{ij} + t_{hk}) + \sigma_1^2 t_{ij} t_{hk} & \text{if } i = h \text{ and } j \neq k \\ 0 & \text{if } i \neq h \end{cases} \tag{2.19}$$

Note that, in some situations, any two observations from two different components of the same system may exhibit some degree of correlation. However, the current literature (e.g., [60, 61, 94, 98, 148]) plainly assumes that the degradation of different components in nuclear power plants is independent. This assumption produces fairly accurate results with the advantage of modeling and computational convenience.

Likelihood Function

Oftentimes, it is assumed that the measurement noise is generated only from inspection tools (e.g., ultrasonic probes in nuclear power plants). In this scenario, a simple and

plausible assumption can be made about the noise term, i.e., it follows the normal distribution. This assumption also brings analytical convenience to the parameter estimation process by making noisy likelihoods analytically tractable. For instance, given $z_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_z^2)$, the noisy likelihood can be easily derived to be a product of multi-dimensional normal distributions [140], i.e.,

$$\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\Sigma}_b, \sigma_z^2 \mid \mathbf{y}_1, \cdots, \mathbf{y}_N) = \prod_{i=1}^{N} (2\pi)^{-m_i/2} |\boldsymbol{\Sigma}_i|^{-1/2} \exp\left[-\frac{1}{2} (\mathbf{y}_i - \mathbf{T}_i\boldsymbol{\beta})^\top \boldsymbol{\Sigma}_i^{-1} (\mathbf{y}_i - \mathbf{T}_i\boldsymbol{\beta}) \right] \tag{2.20}$$

where $\boldsymbol{\beta} = \{\beta_0, \beta_1\}^\top$, $\mathbf{y}_i = \{y_{i1}, y_{i2}, \cdots, y_{im_i}\}^\top$, $\boldsymbol{\Sigma}_i = \mathbf{T}_i \boldsymbol{\Sigma}_b \mathbf{T}_i^\top + \sigma_z^2 \mathbf{I}_{m_i}$, and

$$\mathbf{T}_i^\top = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ t_{i1} & t_{i2} & \cdots & t_{im_i} \end{bmatrix} \tag{2.21}$$

Here $\mathbf{I}_{m_i}$ is an m_i-dimensional unit diagonal matrix. Using the noisy likelihood in Equation 2.20, Bayesian inference of the regression parameters is quite easy since we have the advantage of using the standard MCMC method – the Gibbs sampler [140].

In real-life problems, the noise may come from different sources, such as human error (spatially unreferenced grids), coverage issues, and the probe signal loss [69]. In this situation, the normal distribution may not represent the noise accurately. One may wish to model the noise term using a non-normal mixture of distributions. This, however, leads to a major problem: the regression model produces an intractable likelihood that is difficult to compute during the process of parameter estimation. For instance, suppose the error terms $\{z_{ij}\}$ are modeled as iid random variables with the distribution $f(z) = \sum_{i=1}^{n} w_i f_i(z)$, where $w_i > 0$, $i = 1, \cdots, n$, are weights, and $\{f_i(z)\}$, $i = 1, \cdots, n$, is a set of finite number of distributions. The model likelihood, in this

case, can be written as

$$\mathcal{L} = \prod_{i=1}^{N} \int_{\mathbf{z}_i} f(\mathbf{y}_i - \mathbf{T}_i\boldsymbol{\beta} - \mathbf{z}_i \mid \boldsymbol{\Sigma}_i')f(\mathbf{z}_i)d\mathbf{z}_i \tag{2.22}$$

where $\boldsymbol{\Sigma}_i' = \mathbf{T}_i\boldsymbol{\Sigma}_b\mathbf{T}_i^\top$, $\mathbf{z}_i = \{z_{i1}, z_{i2}, \cdots, z_{im_i}\}^\top$, and $f(\bullet \mid \boldsymbol{\Sigma}_i')$ is normally distributed with zero mean and covariance matrix $\boldsymbol{\Sigma}_i'$. The likelihood function in Equation 2.22 is a high-dimensional integration that is not only analytically intractable but also computationally expensive to evaluate.

2.3 Flaw Generation Model

2.3.1 Non-Homogeneous Poisson Process Model

The previously mentioned random variable, gamma process, and LMER models are defined on continuous sample spaces; thus, suitable to characterize the flaw growth phenomena. However, characterizing flaw generation needs stochastic processes that can model and predict the number of occurrences of flaws in a component. Thus, the counting process models are suitable for modeling flaw generation. The Poisson process is the most popular counting process model and its application in localized corrosion modeling, such as pitting corrosion, can be found in several studies in the literature [38, 63, 109, 137, 148]. For example, Hong [63] used the Poisson process to model pit generation and a Markov process to model the pit depth. The authors derived the distribution of corrosion pit depth and the probability of time-to-failure using their proposed model. Similarly, Valor et al. [137] proposed a new model for simulating pit generation and growth that is based on the NHPP and non-homogeneous Markov process. The authors claimed that their proposed model can satisfactorily reproduce experimental observations and works better than the models available in the literature.

Datla et al. [38] proposed an NHPP model for modeling pit generation in steam generator tubes, whereas for modeling the distribution of the peaks over threshold for pit depths, the authors proposed a generalized Pareto distribution.

Basic properties

A continuous-time stochastic process $N(t)$ is called a non-homogeneous Poisson process with a power law intensity function $\nu(t) = \lambda\delta t^{\delta-1}$, if it has the following properties [34]:

1. At time $t = 0$, $N(t) = 0$.

2. For $0 \leq t_1 \leq t_2 \leq \cdots \leq t_n$, all the increments of $N(t)$, i.e., $N(t_1) - N(0), N(t_2) - N(t_1), \cdots, N(t_n) - N(t_{n-1})$ are independent random variables.

3. All increments $N_i = N(t_i) - N(t_{i-1})$, $i = 1, 2, \cdots, k$, follow the Poisson distribution as

$$\mathbb{P}[N_i = n] = \frac{[\Lambda(t_i, t_{i-1})]^n}{n!} \exp[-\Lambda(t_i, t_{i-1})],\ n = 0, 1, 2, \cdots \tag{2.23}$$

   where $\Lambda(t_i, t_{i-1}) = \int_{t_{i-1}}^{t_i} \nu(t)dt = \lambda(t_i^\delta - t_{i-1}^\delta)$ and $\mathbb{P}[\bullet]$ represents the probability of an event.

The intensity function becomes constant with time when δ is set to 1. The process is then called a homogeneous Poisson process which has the following mean and variance

$$\mathbb{E}[N(t)] = \lambda t,\ \mathrm{Var}[N(t)] = \lambda t \tag{2.24}$$

The parameter λ represents the average number of occurrences of flaws per unit time. Computer simulation of the Poisson process is quite easy since one can simulate the

inter-arrival times between each occurrences from an exponential distribution with rate λ. Figure 2.4 shows several simulated sample paths of the Poisson process.

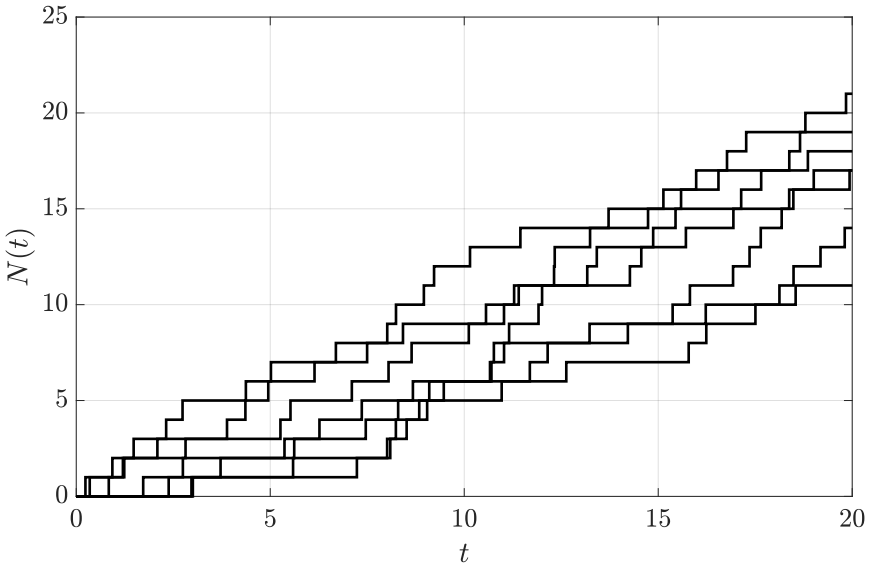


Figure 2.4: Simulated sample paths of the Poisson process model.

NHPP-Weibull Flaw Generation Model

Flaw numbers and flaw sizes, both are included in modeling the flaw generation process, which considers that the flaw generation process is an NHPP and the flaw sizes are iid Weibull random variables with the PDF

$$f_H(h) = \frac{\beta}{\gamma}\left(\frac{h}{\gamma}\right)^{\beta-1} \exp\left[-\left(\frac{h}{\gamma}\right)^{\beta}\right],\ h > 0 \tag{2.25}$$

where $\gamma > 0$ is the scale parameter and $\beta > 0$ is the shape parameter. This model assumes that at each inspection campaign all the previously detected flaws are repaired and only the newly generated flaws are detected; also at time zero, there are no flaws. Another assumption of the model is that flaws grow rapidly to a certain extent and

then they stop or grow at a very slow rate [148].

To characterize the uncertainty in detecting small defects or flaws, the probability of detection (POD) function $p(h)$ is generally used, where h denotes the flaw size. To indicate flaw detection, a binary random variable D is used; where $D = 1$, if the flaw is detected, and $D = 0$, otherwise. Hence, POD can be defined as a probability of having $D = 1$ given the detected flaw size, i.e., $p(h) = \mathbb{P}[D = 1 \mid H = h]$ – a conditional probability function. The POD function, considered by Yuan et al. [148], is adopted in this study:

$$p(h) = \begin{cases} 1 - \dfrac{1 + e^{-qw}}{1 + e^{q(h-w-t_h)}}, & \text{if } h > t_h \\ 0, & \text{otherwise} \end{cases} \tag{2.26}$$

where, w, q and t_h are POD model parameters. Here, t_h is the detection threshold, i.e, a flaw having size less than t_h will not be detected. To control the overall detection quality of the POD function, the other two parameters are used [148].

Likelihood Function

Suppose, a total of k inspections are performed at times $t_1, t_2, \cdots, t_k$ to detect the number of flaws generated in a component. Because of the imperfect detectability, the number of flaws detected at the ith inspection is denoted as n_{di}, $i = 1, 2, \cdots, k$, and the number of undetected flaws are represented using n_{ui}. Thus, the true number of flaws generated between $(i-1)$th and ith inspections are $n_i = n_{di} + n_{ui}$. The true flaw sizes of the detected flaws at ith inspection can be represented as $\mathbf{h}_i = \{h_{i1}, h_{i2}, \cdots, h_{i,n_{di}}\}^\top$, whereas the measured flaw sizes are denoted as $\mathbf{h}_i^{(m)} = \mathbf{h}_i + \mathbf{z}_i$; where $\mathbf{z}_i = \{z_{i1}, z_{i2}, \cdots, z_{i,n_{di}}\}^\top$ is a vector of iid measurement errors.

The model parameters $\boldsymbol{\Theta} = \{\lambda, \delta, \gamma, \beta\}^\top$ are to be estimated. The sample likelihood of the parameter vector $\boldsymbol{\Theta}$ given the degradation measurements of ith inspection can

be written as

$$\ell_i(\boldsymbol{\Theta} \mid n_{di}, \mathbf{h}_i^{(m)}) = f_{\mathbf{H}_i^{(m)}}(\mathbf{h}_i^{(m)})\mathbb{P}[N_{di} = n_{di}] \tag{2.27}$$

where $\mathbf{h}_i^{(m)}$ is a realization of the random variable $\mathbf{H}_i^{(m)}$, $f_{\mathbf{H}_i^{(m)}}(\mathbf{h}_i^{(m)})$ is the joint density of $\mathbf{H}_i^{(m)}$, n_{di} is a realization of the random variable N_{di}, and $\mathbb{P}[N_{di} = n_{di}]$ is the probability of N_{di} being equal to the number of detected flaws in ith inspection campaign. Since the actual flaw sizes and measurement errors are independent, one can write (see Section A.2 for the derivation)

$$\begin{aligned} f_{\mathbf{H}_i^{(m)}}(\mathbf{h}_i^{(m)}) &= f_{\mathbf{H}_i^{(m)}}(h_{i1}^{(m)}, h_{i2}^{(m)}, \cdots, h_{i,n_{di}}^{(m)}) \\ &= \prod_{j=1}^{n_{di}} f_{H_{ij}^{(m)}}(h_{ij}^{(m)}) \\ &= \prod_{j=1}^{n_{di}} \left\{ \frac{1}{\mathbb{E}[p(h)]} \int_0^\infty p(s) f_H(s) f_Z(h_{ij}^{(m)} - s) ds \right\} \end{aligned} \tag{2.28}$$

where $f_Z(\bullet)$ is the PDF of the iid measurement errors and

$$\mathbb{E}[p(h)] = \int_0^\infty p(h) f_H(h) dh \tag{2.29}$$

On the other hand, the probability of the number of detected flaws can be calculated as (see Section A.3 for the derivation)

$$\mathbb{P}[N_{di} = n_{di}] = \sum_{n_i=0}^{\infty} \frac{\{\Lambda(t_i, t_{i-1})\}^{n_i}}{(n_i - n_{di})! n_{di}!} \exp\{-\Lambda(t_i, t_{i-1})\}\{\mathbb{E}[p(h)]\}^{n_{di}}\{1 - \mathbb{E}[p(h)]\}^{n_i - n_{di}} \tag{2.30}$$

Substituting Equation 2.28 and Equation 2.30 into Equation 2.27, one can obtain the sample likelihood of $\boldsymbol{\Theta}$ from only the ith inspection data. Now, the sample likelihood considering data from all $i = 1$ to k inspection campaigns can be calculated by taking products of the sample likelihoods generated using degradation measurements

from each inspection campaign, i.e.,

$$
\begin{aligned}
&\mathcal{L}(\boldsymbol{\Theta} \mid n_{d1}, \mathbf{h}_1^{(m)}, n_{d2}, \mathbf{h}_2^{(m)}, \cdots, n_{dk}, \mathbf{h}_k^{(m)}) \\
&= \prod_{i=1}^{k} \ell_i(\boldsymbol{\Theta} \mid n_{di}, \mathbf{h}_i^{(m)}) \\
&= \prod_{i=1}^{k} \left[\prod_{j=1}^{n_{di}} \left\{ \frac{1}{\mathbb{E}[p(h)]} \int_0^{\infty} p(s) f_H(s) f_Z(h_{ij}^{(m)} - s) ds \right\} \right. \\
&\qquad \left. \sum_{n_i=0}^{\infty} \frac{\{\Lambda(t_i, t_{i-1})\}^{n_i}}{(n_i - n_{di})! n_{di}!} \exp\{-\Lambda(t_i, t_{i-1})\} \{\mathbb{E}[p(h)]\}^{n_{di}} \{1 - \mathbb{E}[p(h)]\}^{n_i - n_{di}} \right]
\end{aligned}
\tag{2.31}
$$

The likelihood function presented in Equation 2.31 is a very complicated function as it not only contains high-dimensional integrals but also high-dimensional infinite summations.

2.4 Concluding Remarks

This chapter introduced four most-commonly used stochastic models for characterizing different kinds of degradation processes. The basic properties of these models are discussed and the sample likelihoods are derived for degradation data that are subjected to various kinds of inspection errors. It can be observed that the sample likelihoods of the flaw growth models are very high-dimensional integrals, whereas the sample likelihood of the flaw generation model is the product of several one-dimensional integrals and summation of infinite series. Thus, statistical estimation of such models is quite challenging since numerical evaluations of such sample likelihoods are extremely difficult and time-consuming.

Chapter 3

Markov Chain Monte Carlo Methods

3.1 Introduction

This chapter presents a brief introduction to the MCMC methods commonly used for Bayesian inference of model parameters. Bayesian inference is a direct application of the Bayes' theorem. To understand the Bayes' theorem, suppose A and B are two propositions or events. In Bayesian statistics, the probabilities $\mathbb{P}(A)$ and $\mathbb{P}(B)$ are our prior degree of beliefs that the events A and B are true respectively. Then, the updated degree of belief about A being true given B is true can be represented by the conditional probability $\mathbb{P}(A \mid B)$. According to Bayes' theorem, this conditional distribution can be written as,

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(B \mid A)\mathbb{P}(A)}{\mathbb{P}(B)} \tag{3.1}$$

where, $\mathbb{P}(A)$ is called the prior probability of A, and $\mathbb{P}(A \mid B)$ is called the posterior probability of A given B. The Bayesian inference method works in a similar fash-

ion. Suppose the PDF of the observation $\mathbf{D}_{\text{obs}}$ from a probabilistic model is given as $f(\mathbf{D}_{\text{obs}}|\mathbf{\Theta})$, where $\mathbf{\Theta}$ is the unknown parameter vector. First a prior distribution of $\mathbf{\Theta}$, denoted as $f(\mathbf{\Theta})$, is assigned based on the background information, and then, according to the Bayes' theorem, the posterior distribution of $\mathbf{\Theta}$, denoted as $f(\mathbf{\Theta} \mid \mathbf{D}_{\text{obs}})$, is derived using the following expression:

$$f(\mathbf{\Theta} \mid \mathbf{D}_{\text{obs}}) = \frac{f(\mathbf{D}_{\text{obs}} \mid \mathbf{\Theta}) f(\mathbf{\Theta})}{\int_{\mathbf{\Theta}} f(\mathbf{D}_{\text{obs}} \mid \mathbf{\Theta}) f(\mathbf{\Theta}) d\mathbf{\Theta}} = \mathcal{C}\mathcal{L}(\mathbf{\Theta} \mid \mathbf{D}_{\text{obs}}) f(\mathbf{\Theta}) \tag{3.2}$$

where $\mathcal{C} = [\int_{\mathbf{\Theta}} f(\mathbf{D}_{\text{obs}} \mid \mathbf{\Theta}) f(\mathbf{\Theta}) d\mathbf{\Theta}]^{-1}$ is the normalizing constant and $\mathcal{L}(\mathbf{\Theta} \mid \mathbf{D}_{\text{obs}}) = f(\mathbf{D}_{\text{obs}} \mid \mathbf{\Theta})$ is the likelihood function.

The likelihood function $\mathcal{L}(\mathbf{\Theta} \mid \mathbf{D}_{\text{obs}})$ represents the chosen probabilistic model and the information from observed data. The prior distribution $f(\mathbf{\Theta})$ represents all other information that is known or assumed about the model parameter $\mathbf{\Theta}$ other than the observed data. The prior information can be any relevant information regarding the model parameters, such as engineering design data, expert judgment, data from other similar systems, or even lack of information. For more information about the prior distribution, the reader is referred to references [16, 20, 57, 67, 73]. Compared to the classical parameter inference, Bayesian inference is able to incorporate information from sources other than the observed data in a formal way through the prior distribution. Bayesian inference also provides a more natural way for expressing the parameter uncertainty using the posterior distribution of the parameter.

Analytical solutions of the posterior exist only for some simple probabilistic models and specially selected priors (conjugate priors). For most other models, the Bayesian posterior has to be evaluated numerically. Direct numerical evaluation of the Bayesian posterior using Equation 3.2, however, can be quite difficult. First, for some models, such as the degradation models with inspection uncertainties (see Chapter 2), numerical

evaluation of the likelihood function is extremely difficult, making direct calculation of the posterior distribution impractical. In other cases, even if the likelihood function itself is relatively easy to evaluate, calculation of the normalization constant $\mathcal{C}$, which is an integral over the entire admissible region of the model parameters, can still be time-consuming, especially when the parameter vector $\boldsymbol{\Theta}$ is of high dimension.

The computational difficulty of the Bayesian inference method can be overcome using various advanced Monte Carlo methods. Instead of calculating the posterior density numerically, Monte Carlo simulation aims to draw random samples from the posterior distribution. These samples can then be used in subsequent posterior inference or model prediction. In Chapter 3 and Chapter 4, two particular simulation techniques, Markov chain Monte Carlo and approximate Bayesian computation, are introduced. The proposed methods can be used for Bayesian inference of complicated stochastic degradation models subject to inspection uncertainties.

To date, MCMC is the most powerful yet a simple method for generating samples from a distribution using the theory of Markov chains [112]. When direct sampling from a target distribution $f(\bullet)$ is not possible (e.g., $f(\bullet)$ is known only up to a constant of proportionality), MCMC provides an alternate solution by generating Markov chains with $f(\bullet)$ as a stationary distribution. The first MCMC algorithm was proposed by Metropolis et al. [91] in a statistical physics context. Later, Hastings [58] generalized the method as a tool for statistical sampling and proposed the Metropolis-Hastings (MH) algorithm. Among the early applications of MCMC, works by Geman and Geman [52] and Tanner and Wang [125] are notable. Finally, the method was popularized in the Bayesian community by Gelfand and Smith [48]. Detailed discussions on the theoretical and practical backgrounds of the MCMC method can be found in references [6, 22, 41, 53, 55, 111, 113, 122, 127].

Examples of the application of MCMC methods for Bayesian modeling of degradation processes can be found in several literature. For instance, Bousquet et al. [19] used the Bayesian approach for inferring parameters of a gamma process. The authors used the gamma process to model partially observed crack growths and successfully derived estimators for the best maintenance time for industrial components. They performed parameter inference using the MCMC method. Similarly, Zhang and Zhou [149] proposed a new Bayesian dynamic linear model to characterize the growth of corrosion defects on energy pipelines. They used the MCMC simulation method for parameter inference using data from multiple high-resolution in-line inspections. Yuan et al. [148] proposed a Bayesian approach for modeling and predicting the pitting flaws in steam generator tubes to account for the inherent variability involved in the corrosion process as well as in the detecting and sizing uncertainties associated with the inspection tool. In their study, they confirmed that without considering the probability-of-detection issues and measurement errors, the leakage risk resulting from pitting corrosion would be under-estimated, despite the fact that the actual pit depth would usually be over-estimated. The authors used a modified MCMC method for parameter inference that runs data augmentation at each iteration of the algorithm. Similarly, using MCMC with data augmentation, Qin et al. [109] proposed a Bayesian framework for parameter inference of the stochastic corrosion model used in their study to characterize generation and growth of corrosion defects in energy pipelines.

3.2 Monte Carlo Methods in General

The basic idea of Monte Carlo simulation is to draw samples from a target probability density $f(\mathbf{x})$ defined on a multi-dimensional space. These samples are mainly used to

numerically evaluate expectations of the form

$$\mathbb{E}[g(\mathbf{X})] = \int_{\mathbf{X}} g(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \tag{3.3}$$

An unbiased estimate of the above-mentioned integral can be computed as

$$\mathbb{E}^{(n)}[g(\mathbf{X})] = \frac{1}{n} \sum_{i=1}^{n} g(\mathbf{x}^{(i)}) \tag{3.4}$$

where $\{\mathbf{x}^{(i)}\}_{i=1}^{n}$ are the Monte Carlo samples drawn from $f(\mathbf{x})$. According to the strong law of large numbers, $\mathbb{E}^{(n)}[g(\mathbf{X})]$ will almost surely (a.s.) converge to $\mathbb{E}[g(\mathbf{X})]$ as the number of samples goes to infinity [54], i.e.,

$$\mathbb{E}^{(n)}[g(\mathbf{X})] \xrightarrow{a.s.} \mathbb{E}[g(\mathbf{X})], \;\; n \longrightarrow \infty \tag{3.5}$$

Likewise, if $\text{Var}[g(\mathbf{X})] = \sigma_g^2$ is finite, i.e., $\sigma_g^2 < \infty$, then according to the central limit theorem, the estimation error converges to a Gaussian distribution, i.e.,

$$\sqrt{n}\Big\{\mathbb{E}^{(n)}[g(\mathbf{X})] - \mathbb{E}[g(\mathbf{X})]\Big\} \longrightarrow \mathcal{N}(0, \sigma_g^2), \;\; n \longrightarrow \infty \tag{3.6}$$

In the context of probability and statistics, numerical integration using Monte Carlo methods is a superior choice over deterministic schemes because Monte Carlo schemes generate samples from high-probability regions, unlike the latter, which generates samples over the entire integration region, producing most of the function values equal to zero. With the goal to draw samples from $f(\mathbf{x})$, various Monte Carlo samplers can be employed depending on the form of the density function. Suppose $f(\mathbf{x})$ has a standard univariate form of a PDF, say Gaussian or gamma. To generate samples from this PDF, we can simply employ the inverse transform method [115] for a univariate case

based on the computation of the inverse cumulative distribution function (CDF). However, if difficulty in drawing samples increases due to high dimension and non-standard forms of density functions, then more advanced and sophisticated methods, such as rejection sampling, importance sampling, and MCMC are implemented. While rejection sampling and importance sampling methods guarantee to simulate iid samples, MCMC generates a Markov chain of correlated samples.

3.2.1 Rejection Sampling

Rejection sampling can be employed when the target density $f(\mathbf{x})$ is only known up to a constant of proportionality, making direct sampling from it impossible. The basic idea of this algorithm is to draw samples from an alternative easy-to-sample proposal distribution $q(\mathbf{x})$ that satisfies the condition

$$f(\mathbf{x}) \leq Mq(\mathbf{x}), \quad M < \infty \tag{3.7}$$

and accept the proposed samples based on an accept-reject rule [6]. Here, $Mq(\mathbf{x})$ works as an envelope distribution to the target distribution. The implementation steps for the Monte Carlo rejection sampler are presented in Algorithm 1.

Algorithm 1 Rejection sampler

1: **for** $i = 1$ to n **do**
2: **repeat**
3: Generate $\mathbf{x}^*$ from the proposal density $q(\mathbf{x})$.
4: Generate u from a uniform distribution $\mathcal{U}[0, 1]$.
5: Accept $\mathbf{x}^*$ if $u < \frac{f(\mathbf{x}^*)}{Mq(\mathbf{x}^*)}$
6: **until** acceptance
7: set $\mathbf{x}^{(i)} = \mathbf{x}^*$
8: **end for**

Example

Let us take an example to understand how rejection sampling works. This example is adopted from [6]. Suppose, one wants to draw samples from a univariate target distribution $f(x)$ which is known only up to a constant of proportionality:

$$f(x) \propto 0.3\exp(-0.2x^2) + 0.7\exp(-0.2(x-10)^2); \ -\infty < x < \infty \tag{3.8}$$

One can assume that $f(x) = cf'(x)$, where c is the normalizing constant, and $f'(x)$ is the unnormalized target distribution. In this particular case, one can analytically calculate $c = 1/\sqrt{5\pi}$ and compare the result of the rejection sampling with the analytical solution. Let us select a normal proposal density $q(x) = \mathcal{N}(5, 10^2)$. Since c is assumed to be unknown, it is impossible to directly calculate the value of M. However, using the inequality $M/c \geq f'(x)/q(x)$ from Equation 3.7 and a trial and error method, one can find that $M/c = 24$ gives a reasonable solution with a rejection rate of around 80.25% ($c = 1/\sqrt{5\pi}$ gives $M = 6.06$). Figure 3.1 shows the envelope distribution $Mq(x)$, the target distribution $f(x)$, and the unnormalized target distribution $f'(x)$. It can be noticed that the support of the proposal distribution covers the support of the target distribution well. Note that, in problems with high dimensions, one may need to select a very high value of M. which may result in a very high number of rejections. The results from the Monte Carlo rejection algorithm are presented in Figure 3.2. The figure shows that as the number of simulations n is increased, the histogram of the accepted samples reaches closer and closer to the target distribution $f(x)$. To get a closer fit of the target distribution at $n = 10^5$, the algorithm used around 19.75% of the total of 5.06×10^5 samples.

The two major limitations of the rejection sampling method [6] are: (i) bounding $f(\mathbf{x})/q(\mathbf{x})$ with a reasonable constant M is not always possible; and (ii) a large M

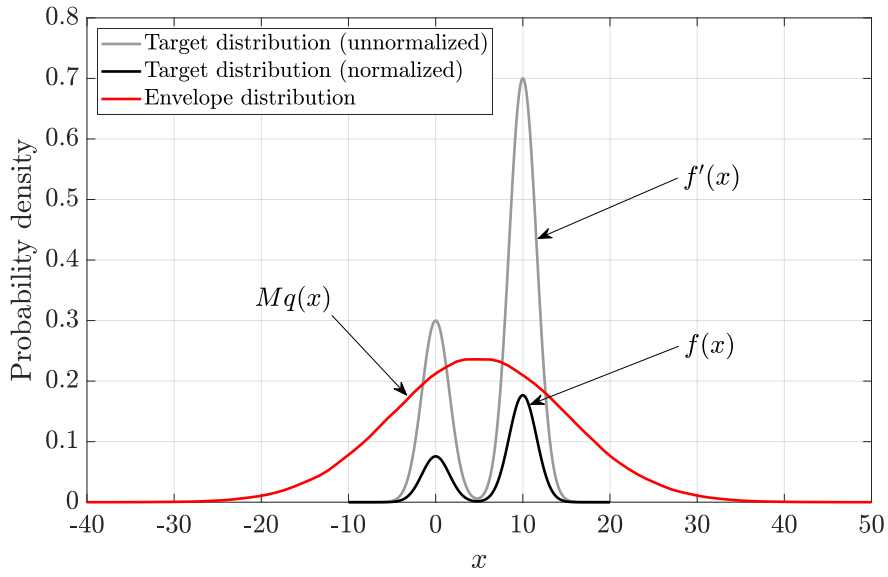


Figure 3.1: The target and envelope distributions used in Monte Carlo rejection sampling.

can produce very small acceptance probabilities (Equation 3.9), making the algorithm computationally prohibitive:

$$\mathbb{P}[\mathbf{x} = \text{acceptance}] = \mathbb{P}\left[u < \frac{f(\mathbf{x})}{Mq(\mathbf{x})}\right] \propto \frac{1}{M} \tag{3.9}$$

Thus, the rejection sampler is deemed impractical for high dimensional problems.

3.2.2 Importance Sampling

Importance sampling is a classical method that is used when direct sampling from $f(\mathbf{x})$ is infeasible. Instead, this method draw samples from an arbitrary importance proposal density $q(\mathbf{x})$ to calculate some numerical estimate and apply a correction to the estimate by multiplying it with the importance weight $w(\mathbf{x}) = f(\mathbf{x})/q(\mathbf{x})$ [6]. The

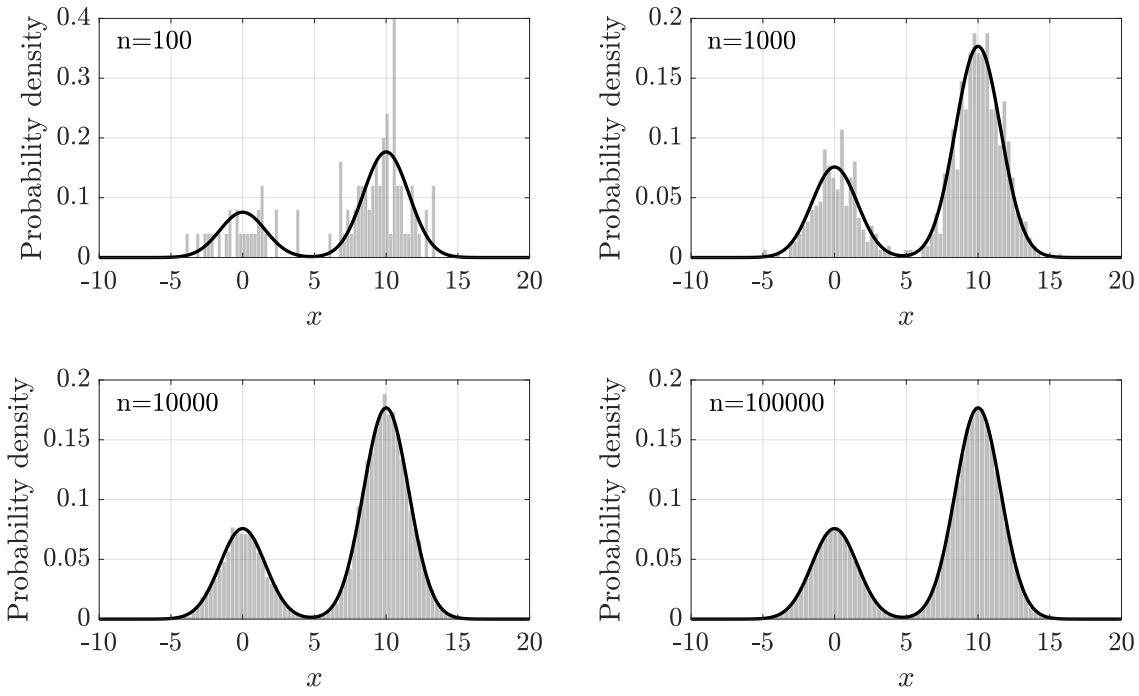


Figure 3.2: Target distribution (black line) and the histogram of samples (in gray) generated using Monte Carlo rejection sampling for different numbers of iterations.

expectation in Equation 3.3 can be re-written as

$$\begin{aligned} \mathbb{E}_f[g(\mathbf{X})] &= \int_{\mathbf{X}} g(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\ \implies \mathbb{E}_q[g(\mathbf{X})w(\mathbf{X})] &= \int_{\mathbf{X}} g(\mathbf{x}) w(\mathbf{x}) q(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{3.10}$$

Consequently, the unbiased Monte Carlo estimator can be written as

$$\mathbb{E}_q^{(n)}[g(\mathbf{X})w(\mathbf{X})] = \frac{1}{n}\sum_{i=1}^{n} g(\mathbf{x}^{(i)}) w(\mathbf{x}^{(i)}) \tag{3.11}$$

Once again, according to the strong law of large numbers, $\mathbb{E}_q^{(n)}[g(\mathbf{X})w(\mathbf{X})]$ will almost surely (a.s.) converge to $\mathbb{E}_q[g(\mathbf{X})w(\mathbf{X})]$ as the number of samples goes to infinity, i.e.,

$$\mathbb{E}_q^{(n)}[g(\mathbf{X})w(\mathbf{X})] \xrightarrow{a.s.} \mathbb{E}_q[g(\mathbf{X})w(\mathbf{X})], \;\; n \longrightarrow \infty \tag{3.12}$$

One can derive the optimal proposal distribution $q^*(\mathbf{x})$ by minimizing the variance of the estimator [6]:

$$q^*(\mathbf{x}) = \frac{|g(\mathbf{x})|f(\mathbf{x})}{\int_{\mathcal{X}} |g(\mathbf{x})|f(\mathbf{x})d\mathbf{x}} \propto |g(\mathbf{x})|f(\mathbf{x}) \tag{3.13}$$

However, usage of the optimal proposal density is not practical since sampling from this density is not easy. The importance sampling name came from the fact that one should target sampling from $f(\mathbf{x})$ in the "important regions" where $|g(\mathbf{x})|f(\mathbf{x})$ is comparatively large – this is the reason behind the high efficiency of the method [6].

Example

Importance sampling is quite useful in estimating small failure probabilities in reliability engineering. As an example, let us assume that $g(x) = (x^2/5 - 32)$ is the performance function of a system, where x follows the probability distribution $f(x)$ presented in Equation 3.8. The failure condition can be defined as $[g(x) > 10]$; the corresponding failure probability P_f is given as

$$P_f = \mathbb{P}[g(x) > 10] = \int_X \mathbb{I}[g(x) > 10]f(x)dx \tag{3.14}$$

where $\mathbb{I}[\bullet]$ is an indicator function:

$$\mathbb{I}[g(x) > 10] = \begin{cases} 1, & \text{if } g(x) > 10 \\ 0, & \text{otherwise} \end{cases}$$

The Monte Carlo estimate of the probability of failure can be written as

$$P_f^{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[g(x^{(i)}) > 10], \quad x^{(i)} \overset{\text{iid}}{\sim} f(x) \tag{3.15}$$

where $x^{(i)}$ can be simulated from $f(x)$ using both rejection and importance sampling methods. After selecting the proposal distribution $q(x)$, P_f can be calculated as

$$P_f^{(n')} = \frac{1}{n'} \sum_{i=1}^{n'} \mathbb{I}[g(x^{(i)}) > 10] w(x^{(i)}), \quad x^{(i)} \overset{\text{iid}}{\sim} q(x) \tag{3.16}$$

where $w(x^{(i)}) = f(x^{(i)})/q(x^{(i)})$. To select the proposal distribution, one needs to identify the "important region". Recall the failure condition:

$$g(x) > 10 \implies \frac{x^2}{5} - 32 > 10 \implies x > 14.4914 \tag{3.17}$$

Clearly, a proposal distribution close to around $x = 14.5$ can make a suitable choice. The proper selection of the variance of a proposal function is very important. It must effectively capture the right tail of the target distribution $f(x)$ since it is the most important region for sampling in failure probability calculation. Let us select a normal distribution as our proposal function, i.e., $q(x) = \mathcal{N}(15, 2^2)$. The target distribution $f(x)$, the failure region, and the proposal distribution $q(x)$ are shown in Figure 3.3.

Figure 3.4 shows the convergence of the probability of failure P_f calculated using both rejection and importance sampling methods. Both methods converged to a value of around $P_f = 1.5 \times 10^{-4}$. However, the figure shows that the iteration or sample size needed to reach convergence for the Monte Carlo rejection sampler is around 4×10^5, whereas the importance sampler converged even before reaching 1×10^5 iterations. This proves that the Monte Carlo importance sampler is far more efficient than the rejection sampler. For importance sampling, it is necessary to choose a proposal density that is easy to simulate and is a good approximation to the target failure region or important region. However, finding this type of proposal density can be challenging in a high-dimensional setting. Thus, more advanced sampling methods, such as Markov chain-based methods, can be used.

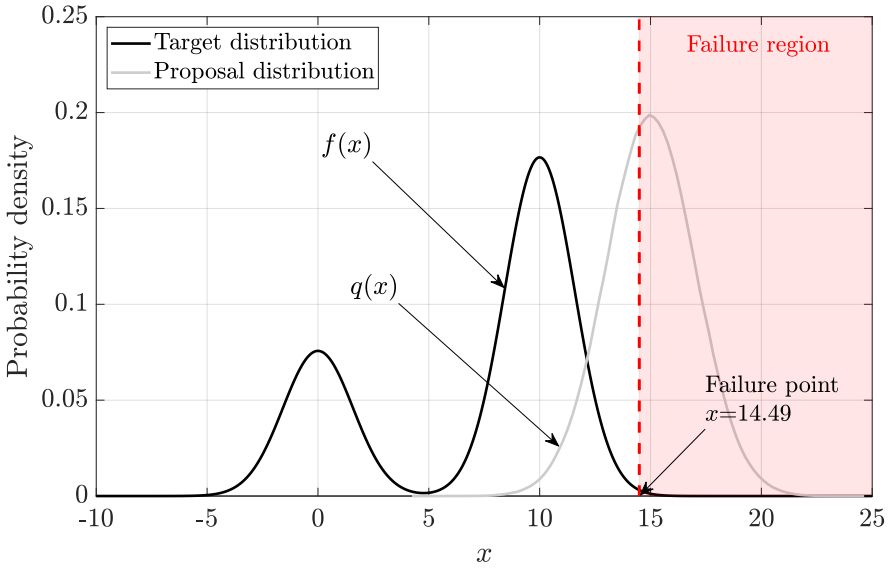


Figure 3.3: Target and proposal distributions along with the failure region.

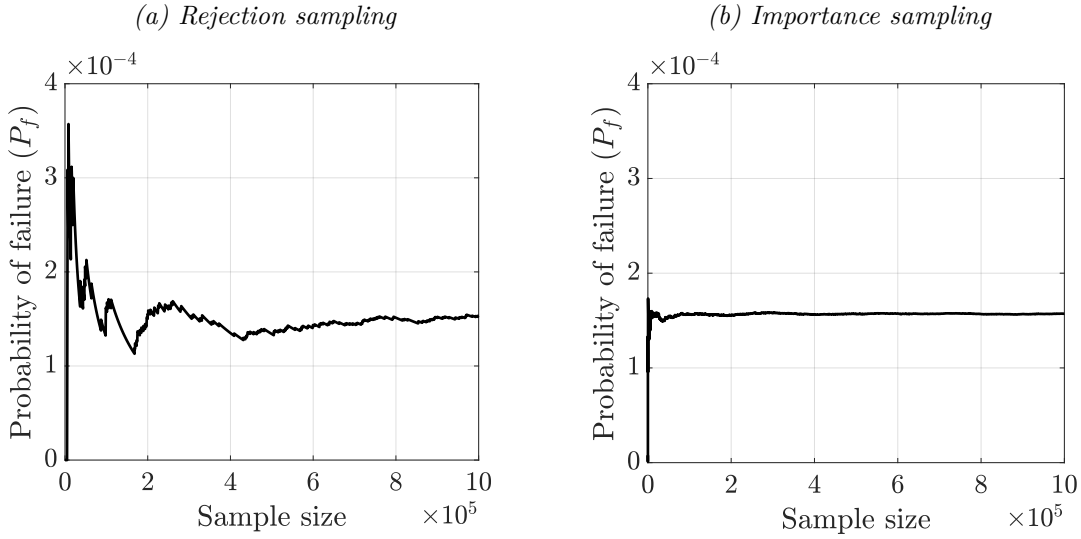


Figure 3.4: Convergence of P_f with respect to the sample size using (a) rejection sampling and (b) importance sampling methods.

3.3 MCMC Algorithms

To understand how MCMC works, one needs to know two of its basic components: the Monte Carlo methods and the Markov chains. The Monte Carlo methods are discussed in Section 3.2. For the basic concepts of Markov chains and related terminologies, the reader is referred to Appendix B. Suppose one needs to draw samples using MCMC from the univariate probability density $f(x)$ as shown in Figure 3.5. The samples

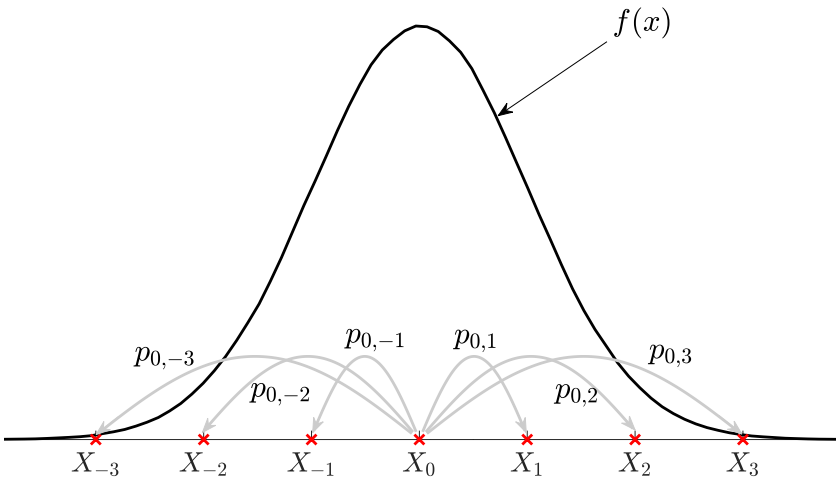


Figure 3.5: A schematic of the target distribution and the finite states of a Markov chain. (http://bjlkeng.github.io)

can be drawn using a finite state Markov chain in conjunction with the Monte Carlo method. In total, there are 7 states, represented as $\{X_{-3}, X_{-2}, \cdots, X_3\}$, and their transition probabilities are assumed to be p_{ij}, $i, j = -3, -2, \cdots, 3$. MCMC tries to spend more time near the high probability region, where $f(x)$ is large, compared to the low probability region, where $f(x)$ is small, in fact, in the exact proportions of $f(x)$. To achieve that, one needs to design the transition probability matrix $\{p_{ij}\}$ in such a way that transitioning from X_0 to other states will have relatively less probability than transitioning from other states to the the central state X_0. Thus, the basic steps

of the MCMC algorithm are:

1. Choose an arbitrary point x.
2. Move to a new point x^* with a transition probability p and stay at the same point with a probability $(1-p)$.
3. Repeat until n number of iterations are completed.

The histogram resulting from those n number of samples should give the target density. However, in reality, each point in the X line is a potential state of the Markov chain, which makes any Markov chain to have countably infinite states. Thus, instead of a transition probability matrix used for a finite state Markov chain, one will have to use a transition kernel to transit from one state to another in the Markov chain that has countably infinite states.

3.3.1 Metropolis-Hastings Algorithm

The MH algorithm [58] is the most widely-known MCMC technique in the scientific community. In fact, all the MCMC samplers can be broadly classified as MH samplers. For instance, the Metropolis [91] and Gibbs samplers [52] can be proved to be special cases of the MH sampler [75], which is a powerful technique to draw samples from distributions known only up to a proportionality constant. For the same reason, the MH algorithm is very popular among the Bayesian statisticians because it allows one to completely avoid the computation of the normalizing constant and draw samples directly from the posterior distribution.

Derivation

To derive the steps of the MH algorithm, one has to make sure that the MCMC sampler generates a Markov chain that sets the target distribution $f(\mathbf{x})$ as its stationary

distribution. To achieve this goal, $f(\mathbf{x})$ needs to satisfy the detailed balance equation (reversibility condition) which can be written as

$$f(\mathbf{x})\mathcal{K}(\mathbf{x}^* \mid \mathbf{x}) = f(\mathbf{x}^*)\mathcal{K}(\mathbf{x} \mid \mathbf{x}^*) \tag{3.18}$$

where $\mathcal{K}(\mathbf{x}^* \mid \mathbf{x})$ is the transition kernel for transitioning from state $\mathbf{x}$ to state $\mathbf{x}^*$. The transition kernel is the equivalent of the transition probability matrix in a continuous state space. The detailed balance equation guarantees that, in a long run, a Markov chain spends equal amounts of time to move from state $\mathbf{x}$ to $\mathbf{x}^*$ and vice versa (for more details, see Section B.4).

One has to calculate the transition kernel from Equation 3.18, which, if exists, will prove that $f(\mathbf{x})$ is indeed the stationary distribution of the Markov chain defined by the same transition kernel. Equation 3.18 can be considered as a basis for constructing any MCMC sampler. Note that, to construct a Markov chain, one has to guarantee that it not only has a stationary distribution but also that is unique. The uniqueness is guaranteed by the ergodicity (irreducibility and aperiodicity) property of the Markov chain (refer to Section B.3 for more details). After re-arranging, Equation 3.18 can be written as

$$\frac{\mathcal{K}(\mathbf{x}^* \mid \mathbf{x})}{\mathcal{K}(\mathbf{x} \mid \mathbf{x}^*)} = \frac{f(\mathbf{x}^*)}{f(\mathbf{x})} \tag{3.19}$$

Note that the ratio in Equation 3.19 guarantees that even if $f(\bullet)$ does not have a standard form but is only known up to a constant of proportionality, it is possible to draw samples from that particular distribution. We can break up the transition kernel $\mathcal{K}(\mathbf{x}^* \mid \mathbf{x})$ into two independent steps: (i) the proposal distribution $q(\mathbf{x}^* \mid \mathbf{x})$, and (ii) the acceptance probability $\mathcal{A}(\mathbf{x}, \mathbf{x}^*)$. By independence, we can write

$$\mathcal{K}(\mathbf{x}^* \mid \mathbf{x}) = q(\mathbf{x}^* \mid \mathbf{x})\mathcal{A}(\mathbf{x}, \mathbf{x}^*) \tag{3.20}$$

The rejection sampling works in a similar fashion, in that the proposal distribution proposes a new sample which is accepted or rejected based on its acceptance probability. One must appropriately choose the proposal distribution $q(\mathbf{x}^* \mid \mathbf{x})$ and derive the acceptance probability $\mathcal{A}(\mathbf{x}, \mathbf{x}^*)$. Since the proposal distribution proposes new points for sampling, if one guarantees that $q(\mathbf{x}^* \mid \mathbf{x})$ has the same support as the target distribution $f(\mathbf{x})$, then the ergodicity condition can be fulfilled [6]. Any standard distribution can be selected as the proposal distribution, and once it is fixed, the next step is to derive the acceptance probability. Substituting Equation 3.20 into Equation 3.19, one can obtain

$$\frac{q(\mathbf{x}^* \mid \mathbf{x})\mathcal{A}(\mathbf{x}, \mathbf{x}^*)}{q(\mathbf{x} \mid \mathbf{x}^*)\mathcal{A}(\mathbf{x}^*, \mathbf{x})} = \frac{f(\mathbf{x}^*)}{f(\mathbf{x})} \implies \frac{\mathcal{A}(\mathbf{x}, \mathbf{x}^*)}{\mathcal{A}(\mathbf{x}^*, \mathbf{x})} = \frac{f(\mathbf{x}^*)q(\mathbf{x} \mid \mathbf{x}^*)}{f(\mathbf{x})q(\mathbf{x}^* \mid \mathbf{x})} \tag{3.21}$$

Thus, one has to choose the acceptance probability in such a way so that it satisfies Equation 3.21. Typically, one can choose

$$\mathcal{A}(\mathbf{x}, \mathbf{x}^*) = \min\left\{1, \frac{f(\mathbf{x}^*)q(\mathbf{x} \mid \mathbf{x}^*)}{f(\mathbf{x})q(\mathbf{x}^* \mid \mathbf{x})}\right\} \tag{3.22}$$

Let us check whether Equation 3.22 satisfies Equation 3.21. Suppose $\frac{f(\mathbf{x}^*)q(\mathbf{x}|\mathbf{x}^*)}{f(\mathbf{x})q(\mathbf{x}^*|\mathbf{x})} \leq 1$ (Case 1), which implies

$$\begin{aligned} \mathcal{A}(\mathbf{x}, \mathbf{x}^*) &= \min\left\{1, \frac{f(\mathbf{x}^*)q(\mathbf{x} \mid \mathbf{x}^*)}{f(\mathbf{x})q(\mathbf{x}^* \mid \mathbf{x})}\right\} = \frac{f(\mathbf{x}^*)q(\mathbf{x} \mid \mathbf{x}^*)}{f(\mathbf{x})q(\mathbf{x}^* \mid \mathbf{x})}, \quad \text{and} \\ \mathcal{A}(\mathbf{x}^*, \mathbf{x}) &= \min\left\{1, \frac{f(\mathbf{x})q(\mathbf{x}^* \mid \mathbf{x})}{f(\mathbf{x}^*)q(\mathbf{x} \mid \mathbf{x}^*)}\right\} = 1 \end{aligned} \tag{3.23}$$

From Equation 3.23, it is clear that the chosen acceptance probability satisfies Equation 3.21 for Case 1.

Now, let us assume that $\frac{f(\mathbf{x}^*)q(\mathbf{x}|\mathbf{x}^*)}{f(\mathbf{x})q(\mathbf{x}^*|\mathbf{x})} > 1$ (Case 2). This implies

$$\begin{aligned}\mathcal{A}(\mathbf{x}, \mathbf{x}^*) &= \min\left\{1, \frac{f(\mathbf{x}^*)q(\mathbf{x} \mid \mathbf{x}^*)}{f(\mathbf{x})q(\mathbf{x}^* \mid \mathbf{x})}\right\} = 1, \quad \text{and} \\ \mathcal{A}(\mathbf{x}^*, \mathbf{x}) &= \min\left\{1, \frac{f(\mathbf{x})q(\mathbf{x}^* \mid \mathbf{x})}{f(\mathbf{x}^*)q(\mathbf{x} \mid \mathbf{x}^*)}\right\} = \frac{f(\mathbf{x})q(\mathbf{x}^* \mid \mathbf{x})}{f(\mathbf{x}^*)q(\mathbf{x} \mid \mathbf{x}^*)}\end{aligned} \tag{3.24}$$

Equation 3.24 proves that, even for Case 2, the chosen acceptance probability satisfies Equation 3.21.

By selecting a proper proposal density and using the acceptance probability given in Equation 3.22, one can develop an algorithm which can construct a Markov chain with the stationary distribution equal to the target distribution $f(\mathbf{x})$. This scheme is known as the popular MH algorithm. The basic implementation steps of the MH sampler [56] are presented in Algorithm 2.

Algorithm 2 MH sampler

1: Initialize by randomly selecting an arbitrary point $\mathbf{x}^{(i)}$, set $i = 1$.
2: **for** $i = 1$ to $n - 1$ **do**
3: Generate $\mathbf{x}^*$ from the proposal distribution $q(\mathbf{x}^* \mid \mathbf{x}^{(i)})$
4: Calculate the acceptance probability

$$\mathcal{A}(\mathbf{x}^{(i)}, \mathbf{x}^*) = \min\left\{1, \frac{f(\mathbf{x}^*)q(\mathbf{x}^{(i)} \mid \mathbf{x}^*)}{f(\mathbf{x}^{(i)})q(\mathbf{x}^* \mid \mathbf{x}^{(i)})}\right\}$$

5: Simulate u from the uniform distribution $\mathcal{U}[0, 1]$.
6: Set

$$\mathbf{x}^{(i+1)} = \begin{cases}\mathbf{x}^*, & \text{if } u \leq \mathcal{A}(\mathbf{x}^{(i)}, \mathbf{x}^*) \\ \mathbf{x}^{(i)}, & \text{otherwise}\end{cases}$$

7: **end for**

Example

Let us implement the MH algorithm to draw samples from the unnormalized target density $f(x) \propto 0.3\exp(-0.2x^2) + 0.7\exp(-0.2(x-10)^2)$, previously presented in Equa-

tion 3.8. The proposal density is selected to be a normal distribution with mean equal to the current sample and standard deviation equal to 3, i.e., $q(x^* \mid x) = \mathcal{N}(x, 3^2)$. The MH sampler generated a Monte Carlo Markov chain of the length of 12000 iterations with an arbitrary initial point $x = 30.73$. The first 2000 samples in the chain are rejected to allow for "burn-in". Burn-in of a Markov chain is a process to discard a few initial samples to allow for the chain to reach a high probability region (for more details, see Section 3.4). The Markov chain and its burn-in region are shown in Figure 3.6. The estimated histograms for different numbers of iterations (i.e., the Markov

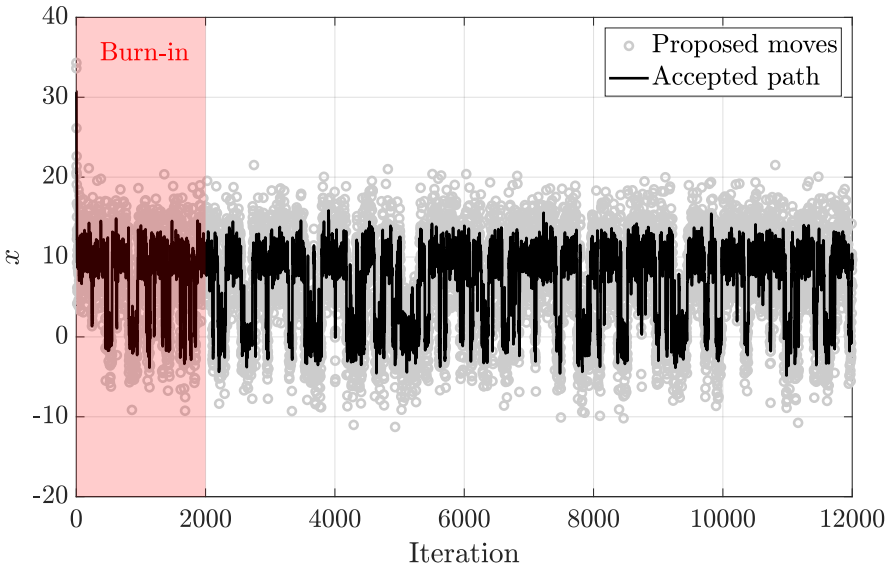


Figure 3.6: Sample iterations generated from the MH sampler.

chain lengths) are presented in Figure 3.7. It can be observed that as the chain length increases, the histogram of the samples cover more and more of the target region, and at $n = 10000$, the histogram shows a great fit with the target distribution.

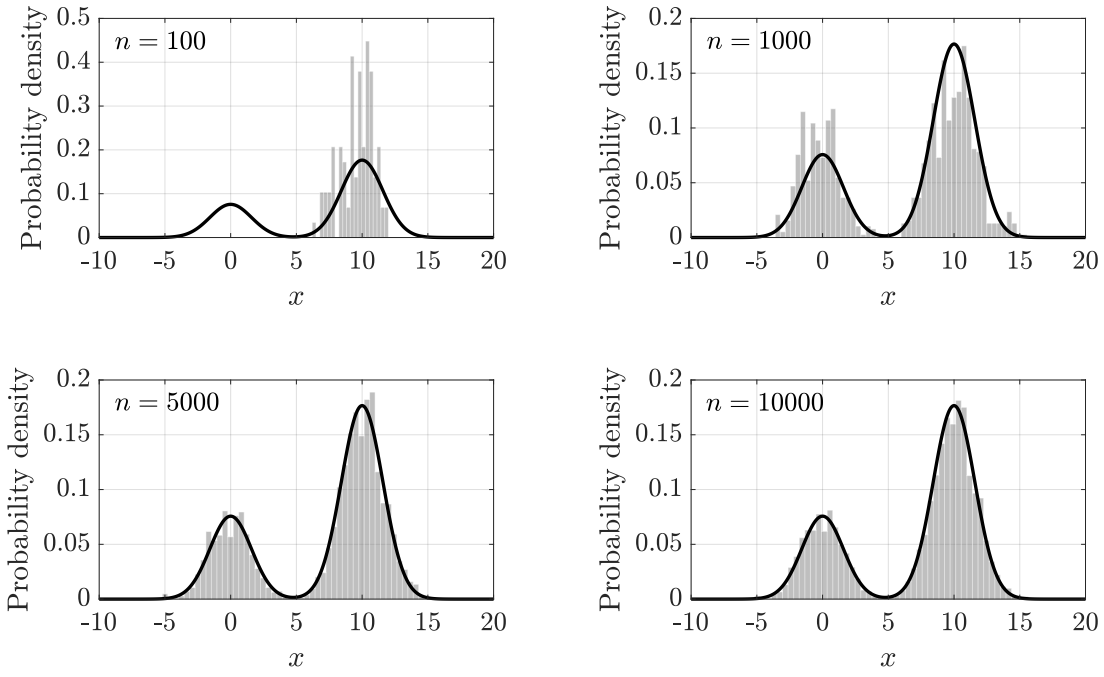


Figure 3.7: Target distribution (black line) and histograms of samples (in gray) for different chain lengths (number of iterations).

3.3.2 Variants of Metropolis-Hastings Algorithm

Depending on the type of the proposal distribution, various simple instances of the generalized MH algorithm (Algorithm 2) can be derived. A few such instances are presented next.

Independent Metropolis-Hastings Algorithm

In the independent sampler, the proposal distribution is independent of the current state, i.e., $q(\mathbf{x}^* \mid \mathbf{x}) = q(\mathbf{x}^*)$. The acceptance probability can be written as [6]

$$\mathcal{A}(\mathbf{x}, \mathbf{x}^*) = \min\left\{1, \frac{f(\mathbf{x}^*)q(\mathbf{x} \mid \mathbf{x}^*)}{f(\mathbf{x})q(\mathbf{x}^* \mid \mathbf{x})}\right\} = \min\left\{1, \frac{f(\mathbf{x}^*)/q(\mathbf{x}^*)}{f(\mathbf{x})/q(\mathbf{x})}\right\} = \min\left\{1, \frac{w(\mathbf{x}^*)}{w(\mathbf{x})}\right\} \tag{3.25}$$

where $w(\mathbf{x}) = f(\mathbf{x})/q(\mathbf{x})$ (similar to the importance weight). Note the close similarity between the independent MH algorithm and the importance sampling algorithm (see Subsection 3.2.2).

Random-Walk Metropolis-Hastings Algorithm

The random walk chain can be constructed following the process $\mathbf{x}^* = \mathbf{x} + \mathbf{z}$, where $\mathbf{z}$ is an iid noise term generated from a proposal density having the form $q(\mathbf{x}^* \mid \mathbf{x}) = q(\mathbf{x}^* - \mathbf{x})$ [113]. Note that if the proposal distribution is symmetric, i.e., $q(\mathbf{x}^* - \mathbf{x}) = q(\mathbf{x} - \mathbf{x}^*)$, then the acceptance probability simplifies to

$$\mathcal{A}(\mathbf{x}, \mathbf{x}^*) = \min\left\{1, \frac{f(\mathbf{x}^*)q(\mathbf{x} \mid \mathbf{x}^*)}{f(\mathbf{x})q(\mathbf{x}^* \mid \mathbf{x})}\right\} = \min\left\{1, \frac{f(\mathbf{x}^*)}{f(\mathbf{x})}\right\} \tag{3.26}$$

The Metropolis algorithm [91] belongs to the same family of symmetric random-walk MH algorithms.

Single-Component Metropolis-Hastings Algorithm

Suppose a random vector $\mathbf{x}$ consists of k components (the components can be multi-dimensional), i.e., $\mathbf{x} = \{x_1, x_2, \cdots, x_k\}^\top$, with the distribution $f(\mathbf{x}) = f(x_1, x_2, \cdots, x_k)$. The objective is to draw samples from the target distribution $f(\mathbf{x})$, which can not be used for direct sampling. Now, according to the single-component MH algorithm, instead of proposing and accepting samples directly according to the basic MH sampler, one can alternatively propose and accept samples component-wise. In other words, instead of proposing a direct move using the proposal distribution $q(\mathbf{x}^* \mid \mathbf{x})$ and accepting the new sample using $\mathcal{A}(\mathbf{x}, \mathbf{x}^*)$, one can propose a component-wise move using $q_j(x_j^* \mid x_j)$ and accept it based on the component-wise acceptance probability [29]

$$\mathcal{A}_j(x_j, x_j^*) = \min\left\{1, \frac{f_j(x_j^*)q_j(x_j \mid x_j^*)}{f_j(x_j)q_j(x_j^* \mid x_j)}\right\} \tag{3.27}$$

where $q_j(x_j^* \mid x_j)$ is dependent on the current value x_j but may or may not depend on the other components $\mathbf{x}_{-j} = \{x_1, x_2, \cdots, x_{j-1}, x_{j+1}, \cdots, x_k\}$, and $f_j(x_j) = f(x_j \mid \mathbf{x}_{-j})$ is the full conditional distribution. If accepted, then x_j will be updated to x_j^*; otherwise, x_j will repeat itself. The process is repeated for $j = 1, 2, \cdots, k$ until the entire vector is updated. The full conditional distributions can be presented as

$$f_j(x_j) = f(x_j \mid \mathbf{x}_{-j}) = \frac{f(\mathbf{x})}{f(\mathbf{x}_{-j})} = \frac{f(\mathbf{x})}{\int f(\mathbf{x}) dx_j} \tag{3.28}$$

According to the product of kernels principle [31, 58], the transition kernel of this algorithm can be written as

$$\mathcal{K}(\mathbf{x}^* \mid \mathbf{x}) = \prod_{j=1}^{k} \mathcal{K}_j(x_j^* \mid x_j) = \prod_{j=1}^{k} q_j(x_j^* \mid x_j) \mathcal{A}_j(x_j, x_j^*) \tag{3.29}$$

where $\mathcal{K}_j(x_j^* \mid x_j)$, $j = 1, 2, \cdots, k$, are the conditional transition kernels (assuming they exist). The steps of the algorithm are presented in Algorithm 3.

Algorithm 3 Single-component MH sampler

1: Initialize $\mathbf{x}^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \cdots, x_k^{(i)}\}$, set $i = 1$.
2: **for** $i = 1$ to $n - 1$ **do**
3: **for** $j = 1$ to k **do**
4: Generate x_j^* from the proposal distribution $q_j(x_j^* \mid x_j^{(i)})$
5: Calculate the acceptance probability

$$\mathcal{A}_j(x_j^{(i)}, x_j^*) = \min\left\{1, \frac{f_j(x_j^*) q_j(x_j^{(i)} \mid x_j^*)}{f_j(x_j^{(i)}) q_j(x_j^* \mid x_j^{(i)})}\right\}$$

6: Simulate u from the uniform distribution $\mathcal{U}[0, 1]$.
7: Set

$$x_j^{(i+1)} = \begin{cases} x_j^*, & \text{if } u \leq \mathcal{A}_j(x_j^{(i)}, x_j^*) \\ x_j^{(i)}, & \text{otherwise} \end{cases}$$

8: **end for**
9: **end for**

Langevin Algorithm

This algorithm is motivated by the idea of discrete approximation of the Langevin diffusion process [113]. Here, the proposal density is given by

$$q(\mathbf{x}^* \mid \mathbf{x}) = \mathcal{N}(\mathbf{x} + (\delta/2)\nabla \log f(\mathbf{x}), \delta) \tag{3.30}$$

where $\delta > 0$ is some small quantity.

3.3.3 Gibbs Sampler

The Gibbs sampler is a special case of the single-component MH sampler, which was developed by Geman and Geman [52] in the context of image processing problems. It is the simplest algorithm to construct a Markov chain. This algorithm is particularly useful when it is possible to draw samples from conditional distributions of a multivariate random variable but difficult to draw samples directly from its joint distribution.

The Gibbs sampler can be described as follows. Suppose, a random vector $\mathbf{x}$ consists of k random variables, i.e., $\mathbf{x} = \{x_1, x_2, \cdots, x_k\}^\top$, with a target distribution $f(\mathbf{x}) = f(x_1, x_2, \cdots, x_k)$. To derive the acceptance probability of the Gibbs sampler, let us assume that the proposal distributions are equal to the full conditional distributions, i.e.,

$$q_j(x_j^* \mid x_j) = f_j(x_j^*) = f(x_j^* \mid \mathbf{x}_{-j}), \quad j = 1, 2, \cdots, k \tag{3.31}$$

where $\mathbf{x}_{-j} = \{x_1, x_2, \cdots, x_{j-1}, x_{j+1}, \cdots, x_k\}$. Now, one can re-write Equation 3.27 as

$$\mathcal{A}_j(x_j, x_j^*) = \min\left\{1, \frac{f_j(x_j^*) q_j(x_j \mid x_j^*)}{f_j(x_j) q_j(x_j^* \mid x_j)}\right\} = \min\left\{1, \frac{f_j(x_j^*) f_j(x_j)}{f_j(x_j) f_j(x_j^*)}\right\} = 1 \tag{3.32}$$

which shows that the acceptance probability in the Gibbs sampler is 1; thus, the proposed candidates are always accepted. The Gibbs sampler has two variants:

1. **The deterministic-scan Gibbs sampler**: The univariate components are updated sequentially. The transition kernel is given as [113]

$$\mathcal{K}(\mathbf{x}^* \mid \mathbf{x}) = \prod_{j=1}^{k} \mathcal{K}_j(x_j^* \mid x_j) = \prod_{j=1}^{k} q_j(x_j^* \mid x_j)\mathcal{A}_j(x_j, x_j^*) = \prod_{j=1}^{k} f_j(x_j^*) \tag{3.33}$$

2. **The random-scan Gibbs sampler**: The univariate components are updated randomly. The transition kernel can be written as [75, 113]

$$\mathcal{K}(\mathbf{x}^* \mid \mathbf{x}) = \sum_{j=1}^{k} \alpha_j \mathcal{K}_j(x_j^* \mid x_j) = \sum_{j=1}^{k} \alpha_j q_j(x_j^* \mid x_j)\mathcal{A}_j(x_j, x_j^*) = \sum_{j=1}^{k} \alpha_j f_j(x_j^*) \tag{3.34}$$

where $0 \leq \alpha_j \leq 1$ is the selection probability and $\sum_{j=1}^{k} \alpha_j = 1$. If the components are selected with equal probabilities, then the above equation simplifies to

$$\mathcal{K}(\mathbf{x}^* \mid \mathbf{x}) = \frac{1}{k} \sum_{j=1}^{k} f_j(x_j^*) \tag{3.35}$$

The deterministic-scan and random-scan Gibbs algorithms are presented in Algorithm 4 and Algorithm 5, respectively.

Example

Let us understand the Gibbs sampling method through an example adopted from [77]. Suppose one needs to draw samples from a bivariate normal distribution $f(x_1, x_2)$ that has zero means, unit standard deviations, and a correlation coefficient of $\rho = 0.5$. The joint distribution can be written as

$$f(x_1, x_2) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left[-\frac{x_1^2 - 2\rho x_1 x_2 + x_2^2}{2(1-\rho^2)}\right] \tag{3.36}$$

Algorithm 4 Deterministic-scan Gibbs sampler

Initialize $\mathbf{x}^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \cdots, x_k^{(i)}\}$, set $i = 1$.
for $i = 1$ to $n - 1$ **do**
    Generate $x_1^{(i+1)} \sim f(x_1 \mid x_2^{(i)}, x_3^{(i)}, \cdots, x_k^{(i)})$
    Generate $x_2^{(i+1)} \sim f(x_2 \mid x_1^{(i+1)}, x_3^{(i)}, \cdots, x_k^{(i)})$
        $\vdots$
    Generate $x_k^{(i+1)} \sim f(x_k \mid x_1^{(i+1)}, x_2^{(i+1)}, \cdots, x_{k-1}^{(i+1)})$
end for

Algorithm 5 Random-scan Gibbs sampler

Initialize $\mathbf{x}^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \cdots, x_k^{(i)}\}$, set $i = 1$.
Choose the selection probabilities $\{\alpha_1, \alpha_2, \cdots, \alpha_k\}$.
for $i = 1$ to $n - 1$ **do**
    Randomly choose $j \in \{1, 2, \cdots, k\}$ with probability α_j.
    Generate $x_j^{(i+1)} \sim f(x_j \mid \mathbf{x}_{-j}^{(i)})$.
end for

To derive the full conditional $f(x_1 \mid x_2)$, one needs to pick those terms which involve only the variable x_1, i.e.,

$$f(x_1 \mid x_2) \propto \exp\left[-\frac{x_1^2 - 2\rho x_1 x_2}{2(1-\rho^2)}\right] \propto \exp\left[-\frac{(x_1 - \rho x_2)^2}{2(1-\rho^2)}\right] \tag{3.37}$$

which is a kernel of the normal distribution [55]. Thus, the full conditional distribution $f(x_1 \mid x_2)$ is a normal distribution $\mathcal{N}(\rho x_2, 1-\rho^2)$ with mean ρx_2 and standard deviation $\sqrt{1-\rho^2}$. Similarly, $f(x_2 \mid x_1)$ can be derived to be $\mathcal{N}(\rho x_1, 1-\rho^2)$.

Both the deterministic-scan and the random-scan (with equal selection probabilities) Gibbs samplers are implemented for drawing samples from $f(x_1, x_2)$. The Markov chains are simulated for 10000 iterations, and the initial 1000 samples are rejected for burn-in. The first 50 samples of two Markov chains generated with different initial points from both algorithms are shown in Figure 3.8. It can be observed that samples from both algorithms quickly converged to the high probability region and covered the target distribution entirely.

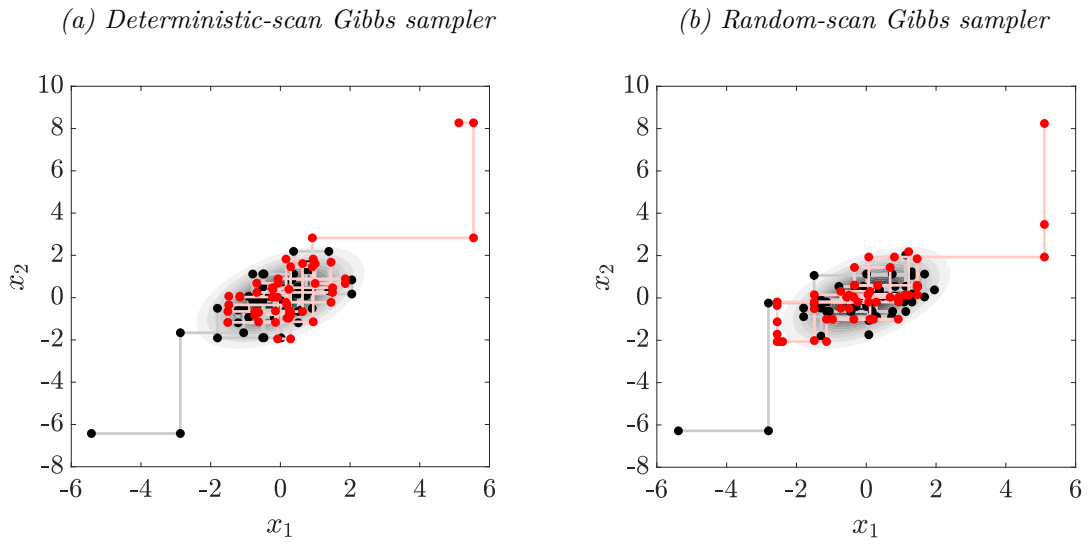


Figure 3.8: The first 50 steps of the (a) deterministic-scan and (b) random-scan Gibbs samplers starting from different initial points.

3.4 Burn-In, Thinning, and Convergence

3.4.1 Burn-In

In the MCMC literature, the "burn-in" of MCMC chains is an important topic. To understand what burn-in is, let us assume that we start our Markov chain from an arbitrary point $\mathbf{x}^{(1)}$. This initial point may belong to a high or low probability region (e.g., tail ends of the target distribution). If the initial point belongs to the high probability region, then the MCMC chain can be assumed to have reached its stationary distribution right from the first iteration. However, if the initialization process starts from a low probability region, then the chain will spend a disproportionate amount of time wandering through the low probability zone giving a false sense of reaching stationarity. The remedy is to throw away some initial samples to allow for the chain to reach a high probability region or burn-in.

The choice of the burn-in period is a complex problem. One can choose this burn-in length by using convergence diagnostics (e.g., [50] or by analytically estimating the appropriate burn-in length in some specific cases (e.g., [92]). However, both these approaches are quite complicated for practical use. In practice, the burn-in length can be chosen in an ad hoc manner [112] by visually inspecting each marginal MCMC chain in case of low dimensional problems. However, in case of high dimensional problems where visual inspection is not possible, a large proportion of the initial samples, say 1% or 2% of the total run, are discarded [53]. If visual inspection is possible, one can also run several chains in parallel, with different initial starting points, and verify whether the chains converge to the same stationary distribution or not after the burn-in period. This step will also help to determine whether or not a chain is trapped inside a high probability region.

3.4.2 Thinning

A major problem with MCMC is the correlation between the two adjacent samples. By definition, the future state of a Markov chain is dependent on the present state. However, the Monte Carlo methods tend to use iid samples to estimate an expectation (or any quantity of interest). Thus, although MCMC draws samples directly from the target distribution, an important property of the Monte Carlo method is lost, i.e., the independence between the generated samples. As a remedy, one can select only the tth, $2t$th, $3t$th, $\cdots$, samples and throw away the rest in between them [53]. If t is large enough, it is possible to get almost independent samples. This process is called thinning of a Markov chain.

One can calculate t approximately by using the sample autocorrelation function (ACF). The sample ACF measures the correlation between $x^{(i+h)}$ and $x^{(i)}$, where h is the lag between the states of a one-dimensional Markov chain. Suppose the burn-in

length of a Markov chain is l. After throwing away l number of initial samples, the remaining number of samples is $n_l = n - l$. According to the reference [21], the most satisfactory estimate of the sample ACF for lag h is given as $r(h) = c(h)/c(0)$, where

$$c(h) = \frac{1}{n_l} \sum_{i=l+1}^{n_l+l-h} (x^{(i+h)} - \bar{x})(x^{(i)} - \bar{x}), \quad 0 < h < n_l \tag{3.38}$$

and $\bar{x} = (1/n_l) \sum_{i=l+1}^{n_l+l} (x^{(i)})$. Note that in case of a high dimensional Markov chain, one can calculate the sample ACF for all one-dimensional marginal chains and verify the correlations between the samples.

3.4.3 Convergence

A very long Markov chain will eventually converge to its target distribution, but how long will that be? Practically, to implement MCMC, one needs to specify a stopping criterion. The stopping criterion not only depends on the problem but also on the quantity that needs to be estimated. Suppose one wants to calculate an expectation, or estimate the parameter uncertainties. In the first case, one can stop the chain when one gets a reasonable value for the estimate of the corresponding expectation. In the latter case, one might need to run the chain long enough to estimate the summary statistics (e.g., mean and standard deviation) of the Markov chain samples. Thus, a stopping criterion can be set by understanding the Monte Carlo uncertainties introduced by the MCMC method and by how much of these uncertainties are acceptable in practice.

Monte Carlo Standard Errors

To calculate the Monte Carlo standard errors, the method of batch means can be employed. This method divides an MCMC chain into equal segments and computes the estimate of interest or the summary statistics of the samples for each segment.

These results show the variability present in the estimates per segment. However, it is important to make sure that the segments are approximately independent by choosing a reasonable segment length. The steps [104] for the method of batch means are presented next.

Suppose $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(n)}$ are the samples from a target density $f(\mathbf{x})$ generated by an MCMC sampler. Now, recall the expectation in the beginning of the chapter, $\mathbb{E}[g(\mathbf{X})] = \int_{\mathbf{X}} g(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$ and its unbiased Monte Carlo estimate $\mathbb{E}^{(n)}[g(\mathbf{X})] = (1/n) \sum_{i=1}^{n} g(\mathbf{x}^{(i)})$. The goal is to calculate the Monte Carlo estimate using several batches of samples. Let's say, the batch size is n_b, where $n/n_b = m$ is an integer. Thus, the chain is divided into the following segments: $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(n_b)}, \mathbf{x}^{(n_b+1)}, \mathbf{x}^{(n_b+2)}, \cdots, \mathbf{x}^{(2n_b)}, \cdots$. The Monte Carlo estimate for the expectation per batch can be computed as

$$b_1 = \frac{1}{n_b} \sum_{i=1}^{n_b} g(\mathbf{x}^{(i)}), b_2 = \frac{1}{n_b} \sum_{i=n_b+1}^{2n_b} g(\mathbf{x}^{(i)}), \ \cdots \ , b_m = \frac{1}{n_b} \sum_{i=(m-1)n_b+1}^{mn_b} g(\mathbf{x}^{(i)}) \tag{3.39}$$

and the batch mean of the estimate can be calculated as

$$\bar{b} = \frac{1}{m} \sum_{j=1}^{m} b_j = \frac{1}{mn_b} \sum_{i=1}^{mn_b} g(\mathbf{x}^{(i)}) \xrightarrow{n \to \infty} \mathbb{E}[g(\mathbf{X})] \tag{3.40}$$

The standard error (Equation 3.6) of the batch mean can be computed assuming that n is very large, i.e.,

$$\sqrt{n}\Big\{\bar{b} - \mathbb{E}[g(\mathbf{X})]\Big\} \longrightarrow \mathcal{N}(0, s^2), \ \ n \longrightarrow \infty \tag{3.41}$$

where s is the batch mean standard error that can be computed as $s = \sqrt{\frac{n_b}{m} \sum_{j=1}^{m} (b_j - \bar{b})^2}$. Note that the Monte Carlo standard error is a quantity with a unit. Thus, it would be difficult to say how small is small enough without understanding the context of the

problem. Hence, a unit-free procedure is more acceptable in general situations. The next subsection presents a different approach which is based on the popular Gelman-Rubin statistic for convergence monitoring of MCMC chains.

Gelman-Rubin Statistic

A unit-free approach for monitoring MCMC convergence relies on the Gelman-Rubin (GR) statistic [50]. Suppose multiple chains are running in parallel with different starting values. Thus, all chains should simultaneously converge to a unique stationary distribution, and after some time, those chains will be indistinguishable in terms of their distribution. One can verify this property by comparing the "variation between chains" to the "variation within chains". However, the analysis is performed in each dimension of the chain separately.

Let us assume that we have in total J number of equal-length one-dimensional chains with different starting values, where $x_j^{(1)}, x_j^{(2)}, \cdots, x_j^{(n)}$ are the samples from the jth chain. The steps [104] for calculating the GR statistic are as follows:

1. Discard l number of samples to allow for Burn-in. The remaining $n_l = n - l$ number of samples, $x_j^{(l+1)}, x_j^{(l+2)}, \cdots, x_j^{(l+n_l)}$, will be used for the assessment.

2. Perform the following calculations:

   2.1. Calculate the within-chain mean $b_j = (1/n_l) \sum_{i=l+1}^{l+n_l} x_j^{(i)}$

   2.2. Calculate the between-chain mean $\bar{b} = (1/J) \sum_{j=1}^{J} b_j$

   2.3. Calculate the between-chain variance $V_b = \{n_l/(J-1)\} \sum_{j=1}^{J} (b_j - \bar{b})$

   2.4. Calculate the within-chain variance $V_w = (1/J) \sum_{j=1}^{J} s_j^2$ where $s_j^2 = \{1/(n_l - 1)\} \sum_{i=l+1}^{l+n_l} (x_j^{(i)} - b_j)^2$

3. Finally, calculate the GR statistic

$$\hat{R} = \frac{\frac{n_l - 1}{n_l} V_w + \frac{1}{n_l} V_b}{V_w} \tag{3.42}$$

Note that if $n_l \to \infty$, $\hat{R} \to 1$, which implies that if one runs the chain for very long, $\hat{R}$ converges to 1. Thus, for practical purposes, one can choose a value of $\hat{R}$ close to 1, say 1.1 or 1.2, for stopping the chain. As a stopping criterion in a multi-dimensional chain, one can choose the largest Markov chain length given by the dimension-wise necessary lengths from the analysis.

3.5 Bayesian Inference using MCMC

In Bayesian inference, the random vector $\mathbf{x}$ and the target distribution $f(\mathbf{x})$ for the MCMC samplers are replaced by the parameter vector $\boldsymbol{\Theta}$ and the posterior distribution $f(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$, respectively. Thus, for generating samples from the target posterior distribution, the following acceptance probability, obtained by substituting Equation 3.2, needs to be used instead of the one given in Algorithm 2:

$$\begin{aligned} \mathcal{A}(\boldsymbol{\Theta}^{(i)}, \boldsymbol{\Theta}^*) &= \min\left\{1, \frac{f(\boldsymbol{\Theta}^* \mid \mathbf{D}_{\text{obs}}) q(\boldsymbol{\Theta}^{(i)} \mid \boldsymbol{\Theta}^*)}{f(\boldsymbol{\Theta}^{(i)} \mid \mathbf{D}_{\text{obs}}) q(\boldsymbol{\Theta}^* \mid \boldsymbol{\Theta}^{(i)})}\right\} \\ &= \min\left\{1, \frac{f(\boldsymbol{\Theta}^*) \mathcal{L}(\boldsymbol{\Theta}^* \mid \mathbf{D}_{\text{obs}}) q(\boldsymbol{\Theta}^{(i)} \mid \boldsymbol{\Theta}^*)}{f(\boldsymbol{\Theta}^{(i)}) \mathcal{L}(\boldsymbol{\Theta}^{(i)} \mid \mathbf{D}_{\text{obs}}) q(\boldsymbol{\Theta}^* \mid \boldsymbol{\Theta}^{(i)})}\right\} \end{aligned} \tag{3.43}$$

It can be observed that the normalizing constant (see Equation 3.2) cancels out since the acceptance probability in the MH sampler is essentially a ratio. Thus, there is no need to explicitly evaluate the normalizing constant for sampling from a target posterior distribution.

Example

Let us apply of the MH algorithm for Bayesian inference using the same example presented in Section 3.3.3. Suppose, the following is an observed data set:

$$\mathbf{D}_{\text{obs}} = \begin{pmatrix} x_{1_{\text{obs}}}^{(1)} & x_{1_{\text{obs}}}^{(2)} & \cdots & x_{1_{\text{obs}}}^{(N)} \\ x_{2_{\text{obs}}}^{(1)} & x_{2_{\text{obs}}}^{(2)} & \cdots & x_{2_{\text{obs}}}^{(N)} \end{pmatrix} \tag{3.44}$$

which is simulated from the the bivariate distribution $f(x_1, x_2)$ (Equation 3.36) with the correlation coefficient $\rho = 0.3$. The number of data points is $N = 100$. Assuming ρ is unknown, the objective is to infer the posterior distribution of ρ, i.e., $f(\rho \mid \mathbf{D}_{\text{obs}}) \propto \mathcal{L}(\rho \mid \mathbf{D}_{\text{obs}}) f(\rho)$. Since ρ varies between -1 and 1, let us select a uniform prior $f(\rho) = \mathcal{U}[-1, 1]$, which makes every value between -1 and 1 to be equally likely for the correlation coefficient. The likelihood function can be written as

$$\begin{aligned} \mathcal{L}(\rho \mid \mathbf{D}_{\text{obs}}) &= \prod_{i=1}^{N} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left[-\frac{(x_{1_{\text{obs}}}^{(i)})^2 - 2\rho(x_{1_{\text{obs}}}^{(i)})(x_{2_{\text{obs}}}^{(i)}) + (x_{2_{\text{obs}}}^{(i)})^2}{2(1-\rho^2)} \right] \\ &= (2\pi)^{-N}(1-\rho^2)^{-N/2} \exp\left[-\frac{\sum_{i=1}^{N}(x_{1_{\text{obs}}}^{(i)})^2 - 2\rho\sum_{i=1}^{N} x_{1_{\text{obs}}}^{(i)} x_{2_{\text{obs}}}^{(i)} + \sum_{i=1}^{N}(x_{2_{\text{obs}}}^{(i)})^2}{2(1-\rho^2)} \right] \end{aligned} \tag{3.45}$$

where N is the number of data vectors. Thus, the posterior distribution can be calculated as

$$\begin{aligned} f(\rho \mid \mathbf{D}_{\text{obs}}) &\propto \prod_{i=1}^{N=2} \frac{1}{\sqrt{1-\rho^2}} \exp\left[-\frac{(x_{1_{\text{obs}}}^{(i)})^2 - 2\rho(x_{1_{\text{obs}}}^{(i)})(x_{2_{\text{obs}}}^{(i)}) + (x_{2_{\text{obs}}}^{(i)})^2}{2(1-\rho^2)} \right] \\ &\propto (1-\rho^2)^{-N/2} \exp\left[-\frac{\sum_{i=1}^{N}(x_{1_{\text{obs}}}^{(i)})^2 - 2\rho\sum_{i=1}^{N} x_{1_{\text{obs}}}^{(i)} x_{2_{\text{obs}}}^{(i)} + \sum_{i=1}^{N}(x_{2_{\text{obs}}}^{(i)})^2}{2(1-\rho^2)} \right] \end{aligned} \tag{3.46}$$

The posterior distribution in Equation 3.46 is not normalized and does not follow any known form. Thus, one cannot draw samples from the posterior directly. However,

the MH algorithm can be implemented to perform the sampling task. In this particular case, one can numerically calculate the normalizing constant of the posterior distribution in Equation 3.46 and verify the results with the analytical posterior distribution. Let us select a normal proposal density to simulate a symmetric random-walk within the MH algorithm: $q(\rho^* \mid \rho) = \mathcal{N}(\rho, 0.2^2)$

Two Markov chains with different initial points, and chain lengths of 10000 iterations are simulated. Figure 3.9 shows that both chains converged quickly to their stationary distributions. After discarding 1000 initial samples for burn-in, the histograms of the posterior samples are generated and presented in Figure 3.9. For comparison,

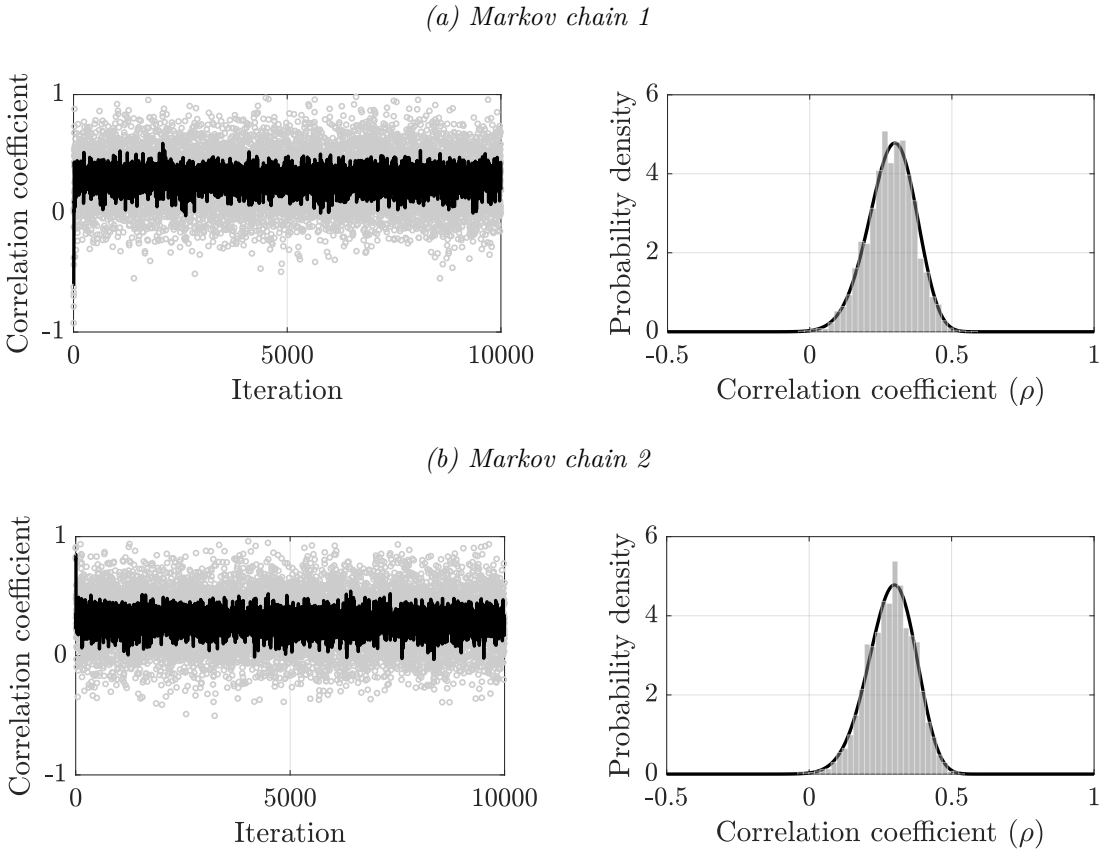


Figure 3.9: Markov chains and corresponding histograms of posterior samples generated from two independent runs of the MH sampler.

the analytically derived target distributions are shown on top of the histograms. It can be observed that both histograms match the target posterior distribution quite well.

3.6 Concluding Remarks

MCMC is a popular choice for Bayesian inference because MCMC samplers can draw samples from complicated posterior distributions. This chapter started with a thorough discussion on the Monte Carlo methods and presented popular MCMC samplers such as the MH algorithm and the Gibbs sampler. To demonstrate how these algorithms work, various numerical examples are included.

The most common Bayesian inference method for estimating parameters is the Gibbs sampler. However, implementing the Gibbs sampler is not always feasible, because: (1) derivations of the individual conditional posterior distributions are not always possible; and (2) in some cases, sampling from the conditional posterior distributions is challenging. Thus, the Gibbs sampler can not be considered a practical approach. On the other hand, a successful implementation of the popular MH sampler depends on the efficiency of evaluating the likelihood function. If the likelihood function is formed using noisy data, such as the noisy likelihoods of the stochastic degradation models in Chapter 2, its numerical evaluation becomes challenging, and this makes the MH algorithm computationally prohibitive. Thus, for Bayesian parameter inference of stochastic degradation models, the evaluation of the likelihood function needs to be circumvented. The next chapter presents various likelihood-free Monte Carlo methods for Bayesian computation, and also discusses how these MCMC sampling algorithms can be used to draw samples from a target posterior distribution without using the likelihood function.

Chapter 4

Approximate Bayesian Computation Methods

4.1 Introduction

The approximate Bayesian computation method [82] is a popular likelihood-free Bayesian inference scheme. Based on a predefined distance function, ABC directly generates samples from the target posterior distribution by comparing the observed data set with numerous simulated data sets. The algorithm employs an "accept-reject" mechanism through a tolerance threshold on the distance values, and retains the relevant parameter samples that satisfy the acceptance criterion. ABC is particularly useful when the model likelihood is intractable or computationally expensive to evaluate. In addition, if data simulation from a forward model is computationally cheap, ABC can achieve reasonable efficiency during the sampling procedure [61].

The idea of the likelihood-free inference was first described by Rubin [114] in 1984. The author presented this method as an intuitive way to understand the Bayesian posterior computation from a frequentist's perspective [82]. Later on Tavaré et al.

[126] used the ABC algorithm as an accept-reject method to infer the genealogy of DNA sequence data in a discrete sample space [82, 124]. Pritchard et al. [108] then extended the ABC method to a continuous sample space and applied it in population genetics problems. Beaumont et al. [12] finally coined the term approximate Bayesian computation in their work related to the Bayesian computation of population genetics. During the last two decades, the development and application of more-sophisticated ABC algorithms in the field of biological sciences (e.g., [10, 43, 83, 120, 128, 130, 134, 145]), engineering (e.g., [3, 4, 27, 30, 32, 59–61, 77, 85, 135]), astronomy (e.g., [5, 66, 68, 144]), archeology (e.g., [36]), psychology (e.g., [133]), geology (e.g., [97]), and hydrology (e.g., [116]) has clearly demonstrated its efficiency and usefulness in Bayesian computation. One can find plenty of review papers on ABC that are available in the literature [11, 37, 82, 110, 124, 133].

The basic ABC method often struggles with complex models that have too many parameters. To solve the parameter estimation problem in complex systems, several well-known and efficient sampling algorithms such as MCMC [83], population Monte Carlo [13], sequential Monte Carlo [130], subset simulation [30], and ellipsoidal nested sampling [4], have been used within the ABC algorithm. A comprehensive list of basic and advanced ABC algorithms available in the literature till date are presented in Table 4.1. The list of abbreviations are provided as a note at the end of the table. The table contains the application areas and the issues found in the existing algorithms.

This chapter provides a brief background of the ABC method and discusses the working principles of various standard ABC algorithms. Moreover, a number of ABC algorithms are developed and the efficacy of those algorithms are demonstrated by comparing them with the standard algorithms in various numerical examples.

Table 4.1: Approximate Bayesian computation algorithms.

Algorithm	**Proposed by**	**Year**	**Application**	**Notes**
ABC-RS (discrete)	Tavaré et al. [126]	1997	Population genetics	Not useful for a continuous sample space.
ABC-RS (continuous)	Pritchard et al. [108]	1999	Population genetics	Very small acceptance probability causes very high rejection rate, making the algorithm highly inefficient
ABC-RS (summary statistics)	Beaumont et al. [12]	2002	Population genetics	(i) Regression adjustment and weighting is proposed; (ii) improves accuracy by reducing bias; (iii) inefficient in high-dimensions.
ABC-MCMC	Marjoram et al. [83]	2003	Population genetics	(i) Efficient compared to ABC-RS; (ii) produce correlated samples; (iv) can get stuck in low probability regions for a very long time.
ABC-PRC	Sisson et al. [120]	2007	Epidemiology	(i) Based on sequential Monte Carlo; (ii) improves efficiency; (iii) induces bias.
ABC-PRC (improved)	Sisson et al. [119]	2008	Population genetics	(i) Improves the original algorithm [120] by proposing a weight update.
ABC-SMC	Toni et al. [130]	2009	Biological dynamical modeling	(i) Based on sequential importance sampling; (ii) improves efficiency; (iii) induces bias.
ABC-PMC	Beaumont et al. [13]	2009	Population genetics	(i) Introduces bias correction to the ABC-PRC algorithms [119, 120] based on genuine importance sampling arguments; (ii) does not require a backward kernel.
Adaptive ABC-SMC	Drovandi and Pettitt [43]	2011	Macro-parasite Population Evolution	(i) Outperforms ABC-SMC [130]; (ii) automatically determines the tolerance sequence and the proposal distribution.
Adaptive ABC-SMC	Del Moral et al. [39]	2012	Disease Epidemiology	(i) More efficient than both the ABC-SMC samplers [43, 130]; (ii) automatically determines the tolerance sequence.
ABC-DE	Turner and Sederberg [132]	2012	Standard toy problems, Psychology	(i) Based on genetic algorithm; (ii) improves efficiency of ABC; (iii) performs extremely well in high-dimensional problems.
ABC-PT	Baragatti et al. [9]	2013	Disease Epidemiology	(i) Based on advanced MCMC method with exchange moves; (ii) does not get trapped locally; (iii) improves accuracy of ABC-MCMC [83].

ABC-SubSim	Chiachio et al. [30]	2014	Structural dynamics	(i) Based on an advanced MCMC method, subset simulation – a rare event simulator; (ii) does not get trapped locally; (iii) more efficient and accurate than ABC-MCMC and ABC-PT.
Noisy ABC	Wilkinson [145]	2014	Population genetics	(i) Properly addresses the problem of model/measurement noise; (ii) produces exact inference results; (iii) small error causes high rejection rate.
Gibbs ABC	Turner and Zandt [134]	2014	Experimental psychology	Suitable for hierarchical Bayesian modeling.
GPS-ABC	Meeds and Welling [86]	2014	Population genetics, Ecology	(i) Stores simulation information in a Gaussian process that acts as a surrogate function for the simulated statistics; (ii) reduce the number of simulations significantly.
HABC	Meeds et al. [87]	2015	Population dynamics	(i) Based on Hamiltonian dynamics; (ii) suitable for high dimensional problems.
Lazy ABC	Prangle [107]	2016	Epidemiology	(i) Based on importance sampling; (ii) Reduced number of model simulations are needed; (iii) Applicable when model simulations are expensive.
K2-ABC	Park et al. [103]	2016	Population dynamics	(i) Utilizes maximum mean discrepancy (MMD) to construct the distance function; (ii) circumvents the need for manually selecting summary statistics.
ABC-NS	Abdessalem et al. [2]	2017	Structural dynamics	(i) Based on an ellipsoidal nested sampling technique; (ii) maintains a relatively high acceptance rate than the traditional ABC algorithms.
ABC-VB	Tran et al. [131]	2017	Standard statistical modeling problems	(i) Based on variational Bayes method; (ii) high accuracy can be achieved.
ABC-QMC	Buchholz and Chopin [23]	2019	Population dynamics, Epidemiology	(i) Based on the quasi Monte Carlo method; (ii) reduces the variance of the posterior.

Note: RS: rejection sampler, MCMC: Markov chain Monte Carlo, PRC: partial rejection control, SMC: sequential Monte Carlo, PMC: population Monte Carlo, DE: differential evolution, PT: parallel tempering, SubSim: subset simulation, GPS: Gaussian process surrogate, HABC: Hamiltonian ABC, K2: kernel embeddings NS: nested sampling, VB: variational Bayes, QMC: quasi Monte Carlo.

4.2 Background

The likelihood-free ABC algorithm directly draw samples from the posterior distribution $f(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) \propto \mathcal{L}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) f(\boldsymbol{\Theta})$ using an accept-reject mechanism. The ABC algorithm starts by generating a candidate parameter set $\{\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Theta}^{(2)}, \cdots\}$ from the prior density $f(\boldsymbol{\Theta})$. Then, with the help of Monte Carlo simulation, ABC simulates the underlying forward model $\mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta})$ multiple times using the candidate parameters, and generates a complete set of corresponding data sets $\{\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, \cdots\}$. Finally, the combinations $\{\boldsymbol{\Theta}^{(k)}, \mathbf{D}^{(k)}\}$ are accepted based on a condition, $\rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}}) \leq \epsilon$, defined by a distance function $\rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}})$ and a tolerance threshold $\epsilon \geq 0$. Depending on the problem, any suitable distance function can be chosen for the rejection mechanism to work. However, to maintain high accuracy, a small tolerance threshold is preferred, which is generally chosen in an ad hoc manner. In the case of large data sets, the distance function can be defined using sufficient summary statistics of the data sets. In this study, the forward model $\mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta})$ could be any degradation model from Chapter 2. A conceptual overview of the likelihood-free ABC method is presented in Figure 4.1.

The ABC method generates samples of $\{\boldsymbol{\Theta}, \mathbf{D}\}$ from the joint distribution $f(\boldsymbol{\Theta}, \mathbf{D} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon)$, which, if marginalized, gives the approximate posterior distribution

$$f(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) \approx f_{\epsilon}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) = \int_{\mathbf{D}} f(\boldsymbol{\Theta}, \mathbf{D} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon) d\mathbf{D} \tag{4.1}$$

where $f_{\epsilon}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$ is called the ABC posterior of $\boldsymbol{\Theta}$. However, there is no need to explicitly evaluate the integration in Equation 4.1 since the samples of $\boldsymbol{\Theta}$ are directly obtained from the samples of the joint posterior $f(\boldsymbol{\Theta}, \mathbf{D} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon)$. The

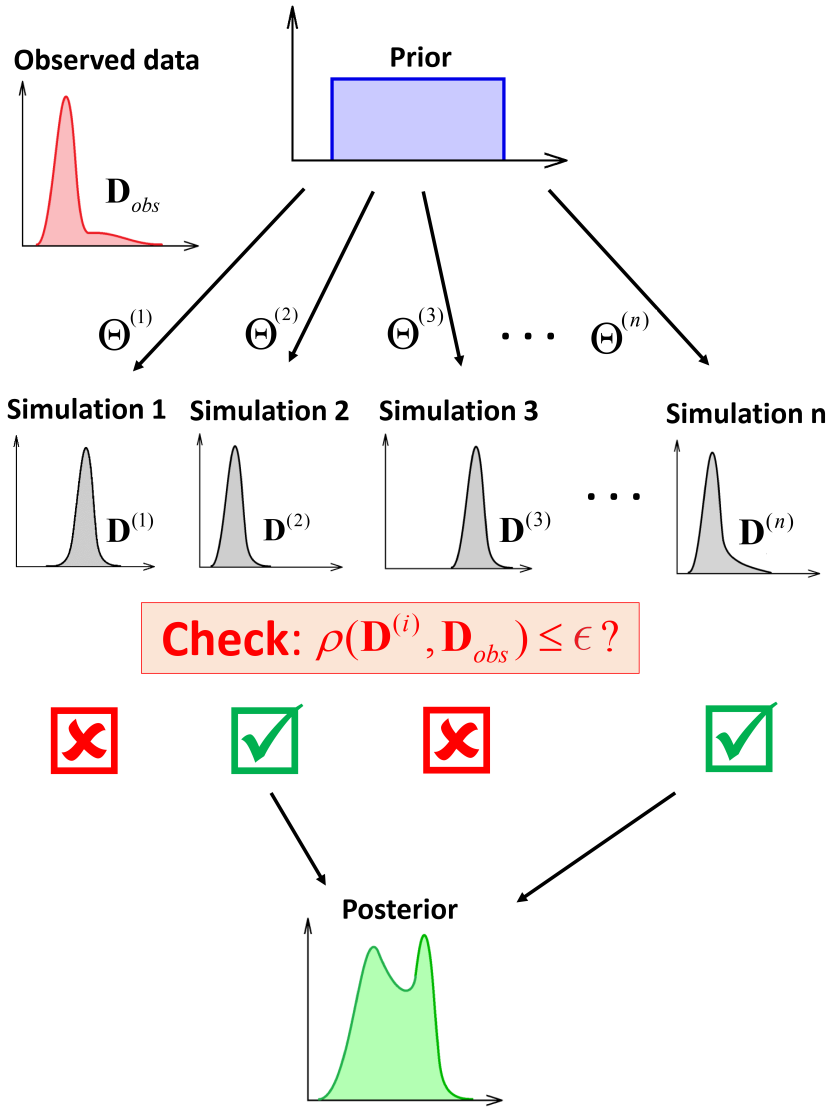


Figure 4.1: A conceptual overview of the ABC method [124].

Bayesian statement for the joint ABC posterior distribution can be written as

$$f(\boldsymbol{\Theta}, \mathbf{D} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon) = \frac{f(\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon \mid \boldsymbol{\Theta}, \mathbf{D}) \mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}) f(\boldsymbol{\Theta})}{\int_{\boldsymbol{\Theta}} \int_{\mathbf{D}} f(\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon \mid \boldsymbol{\Theta}, \mathbf{D}) \mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}) f(\boldsymbol{\Theta}) d\boldsymbol{\Theta} d\mathbf{D}} \tag{4.2}$$

Now, using Equation 4.1, the marginal ABC posterior of $\boldsymbol{\Theta}$ can be calculated as

$$\begin{aligned} f_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) &= \int_{\mathbf{D}} f(\boldsymbol{\Theta}, \mathbf{D} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon) d\mathbf{D} \\ &= \frac{\left\{ \int_{\mathbf{D}} f(\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon \mid \boldsymbol{\Theta}, \mathbf{D}) \mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}) d\mathbf{D} \right\} f(\boldsymbol{\Theta})}{\int_{\boldsymbol{\Theta}} \int_{\mathbf{D}} f(\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon \mid \boldsymbol{\Theta}, \mathbf{D}) \mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}) f(\boldsymbol{\Theta}) d\boldsymbol{\Theta} d\mathbf{D}} \\ &= \frac{\mathcal{L}_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) f(\boldsymbol{\Theta})}{\int_{\boldsymbol{\Theta}} \mathcal{L}_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) f(\boldsymbol{\Theta}) d\boldsymbol{\Theta}} \end{aligned} \tag{4.3}$$

where $\mathcal{L}_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$ is the approximate likelihood function of the ABC algorithm. The ABC likelihood function can be further simplified as

$$\begin{aligned} \mathcal{L}_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) &= \int_{\mathbf{D}} f(\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon \mid \boldsymbol{\Theta}, \mathbf{D}) \mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}) d\mathbf{D} \\ &= \int_{\mathbf{D}} f(\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \mid \boldsymbol{\Theta}, \mathbf{D}) \mathbb{1}\Big[\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon\Big] \mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}) d\mathbf{D} \end{aligned} \tag{4.4}$$

where $\mathbb{1}[\,\bullet\,]$ is an indicator function:

$$\mathbb{1}\Big[\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon\Big] = \begin{cases} 1, & \text{if } \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

It can be noticed that if $\epsilon \to \infty$, the approximate posterior converges to the prior distribution, i.e.,

$$f_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) \xrightarrow{\epsilon \to \infty} f(\boldsymbol{\Theta}) \tag{4.6}$$

By contrast, if the chosen tolerance threshold ϵ is sufficiently small, then the ABC posterior converges to the exact posterior as

$$f_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) \xrightarrow{\epsilon \to 0} f(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) \tag{4.7}$$

However, the selection of a small tolerance threshold often comes with a higher computational cost.

4.3 Standard ABC Algorithms

4.3.1 ABC Rejection Sampler

Following the intuitive idea of ABC, Pritchard et al. [108] proposed the first basic ABC algorithm, also known as the ABC rejection sampler (ABC-RS). The steps of the algorithm are presented in Algorithm 6. The algorithm is based on a similar idea of the Monte Carlo rejection sampler presented in Algorithm 1.

Algorithm 6 ABC rejection sampler

1: **for** $i = 1$ to n **do**
2: **repeat**
3: Generate $\boldsymbol{\Theta}^*$ from the prior $f(\boldsymbol{\Theta})$
4: Simulate a data set $\mathbf{D}^*$ from the model $\mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}^*)$
5: Accept $\boldsymbol{\Theta}^*$ if $\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon$
6: **until** acceptance
7: set $\boldsymbol{\Theta}^{(i)} = \boldsymbol{\Theta}^*$
8: **end for**

Example

The basic ABC algorithm is applied to solve the example problem of inferring the posterior of ρ presented in Section 3.5. To avoid confusion with the distance function $\rho(\bullet, \bullet)$, the notation for the correlation coefficient ρ will be replaced by ρ_c. The same uniform prior $f(\rho_c) = \mathcal{U}[-1, 1]$ is selected for numerical computation. The data generation model is the bivariate distribution $f(x_1, x_2)$ shown in Equation 3.36. For the distance function, a suitable choice would be to compare the sample correlation coefficients of the observed data with that of the simulated data. Thus, the distance function can be calculated as $\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) = |r(\mathbf{D}) - r(\mathbf{D}_{\text{obs}})|$, where $r(\mathbf{D})$ is the sample correlation coefficient [141],

$$r(\mathbf{D}) = \frac{\sum_{i=1}^{N}\left(x_1^{(i)} - \bar{x}_1\right)\left(x_2^{(i)} - \bar{x}_2\right)}{\sqrt{\sum_{i=1}^{N}\left(x_1^{(i)} - \bar{x}_1\right)^2}\sqrt{\sum_{i=1}^{N}\left(x_2^{(i)} - \bar{x}_2\right)^2}} \tag{4.8}$$

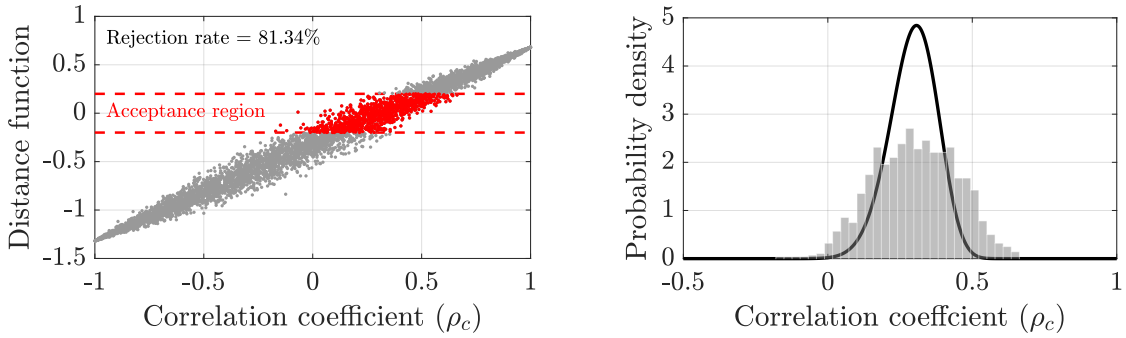
and $\bar{x}_1$ and $\bar{x}_2$ are the sample means. ABC-RS is implemented to draw $n = 1000$ samples using three different values of the tolerance threshold, i.e., $\epsilon = \{0.2, 0.1, 0.005\}$.

Figure 4.2 shows the inference results for different values of the tolerance thresholds. On the left, we have the distance function plotted against the simulated correlation coefficients, whereas on the right, we have the target posterior distribution and the corresponding histograms. It can be noticed that, for a smaller acceptance region, resulting from a smaller tolerance threshold, higher accuracy can be achieved – with correspondingly higher rejection rate. In fact, for the $\epsilon = 0.005$ case, where the ABC posterior almost resembles the true posterior, most of the samples were rejected – a rejection rate of 99.52% is observed. The tolerance threshold can be reduced further to achieve higher accuracy, but doing so will definitely need enormous number of model simulations due to the very high rate of rejection. This tells us that we need a way to reduce the rejection rate by sampling close to the high probability region. This can be achieved by using an MCMC-based ABC method that quickly converges to the high probability region and simulates samples only near the target region.

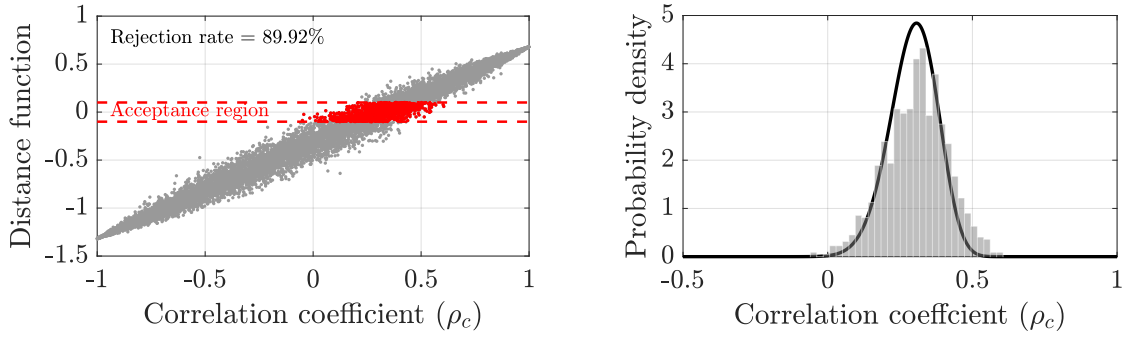
4.3.2 ABC-MCMC Algorithm

Marjoram et al. [83] proposed a new ABC algorithm that uses the MCMC method as its sampling scheme. MCMC proposes new parameter samples from the high probability region and ABC applies its basic accept-reject criterion to obtain the posterior parameter samples. In this way, ABC-MCMC greatly reduces the rejection rate in the ABC algorithm. The steps to implement the ABC-MCMC algorithm are presented

(a) $\epsilon = 0.2$



(b) $\epsilon = 0.1$



(c) $\epsilon = 0.005$

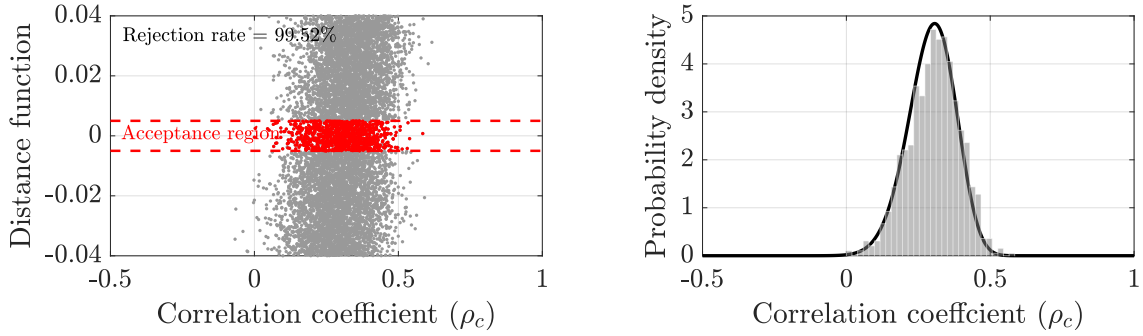


Figure 4.2: Acceptance regions (left), and histograms of posterior samples (right) simulated using ABC-RS for three different tolerance thresholds.

in Algorithm 7. It can be observed that the MCMC scheme used within the ABC algorithm is the popular MH sampler (Algorithm 2).

Algorithm 7 ABC-MCMC sampler

1: Initialize $\boldsymbol{\Theta}^{(i)} \sim f(\boldsymbol{\Theta})$, set $i = 1$.
2: **for** $i = 1$ to $n - 1$ **do**
3: Generate $\boldsymbol{\Theta}^*$ from the proposal distribution $q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(i)})$.
4: Simulate a data set $\mathbf{D}^*$ from the forward model $\mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}^*)$
5: Calculate the distance function $\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}})$.
6: Calculate the acceptance probability

$$\mathcal{A}(\boldsymbol{\Theta}^{(i)}, \boldsymbol{\Theta}^*) = \min\left\{1, \frac{f(\boldsymbol{\Theta}^*)q(\boldsymbol{\Theta}^{(i)} \mid \boldsymbol{\Theta}^*)}{f(\boldsymbol{\Theta}^{(i)})q(\boldsymbol{\Theta}^* \mid \boldsymbol{\Theta}^{(i)})}\right\}$$

7: Simulate u from the uniform distribution $\mathcal{U}[0, 1]$.
8: Set

$$\boldsymbol{\Theta}^{(i+1)} = \begin{cases} \boldsymbol{\Theta}^*, & \text{if } \rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon \text{ and } u \leq \mathcal{A}(\boldsymbol{\Theta}^{(i)}, \boldsymbol{\Theta}^*) \\ \boldsymbol{\Theta}^{(i)}, & \text{otherwise} \end{cases}$$

9: **end for**

To understand how MCMC is integrated within ABC, one needs to establish the connection between the two. The goal of the ABC-MCMC algorithm is to draw samples from the approximate ABC posterior by generating a Markov chain. For the target ABC posterior $f(\boldsymbol{\Theta}, \mathbf{D} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon)$, the acceptance probability can be written as

$$\mathcal{A}(\{\boldsymbol{\Theta}, \mathbf{D}\}, \{\boldsymbol{\Theta}^*, \mathbf{D}^*\}) = \min\left\{1, \frac{f(\boldsymbol{\Theta}^*, \mathbf{D}^* \mid \rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon)q_p(\boldsymbol{\Theta}, \mathbf{D} \mid \boldsymbol{\Theta}^*, \mathbf{D}^*)}{f(\boldsymbol{\Theta}, \mathbf{D} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon)q_p(\boldsymbol{\Theta}^*, \mathbf{D}^* \mid \boldsymbol{\Theta}, \mathbf{D})}\right\}$$

Substituting Equation 4.2, we get

$$\begin{aligned} &= \min\left\{1, \frac{f(\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon \mid \boldsymbol{\Theta}^*, \mathbf{D}^*)\mathcal{M}(\mathbf{D}^* \mid \boldsymbol{\Theta}^*)f(\boldsymbol{\Theta}^*)q_p(\boldsymbol{\Theta}, \mathbf{D} \mid \boldsymbol{\Theta}^*, \mathbf{D}^*)}{f(\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon \mid \boldsymbol{\Theta}, \mathbf{D})\mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta})f(\boldsymbol{\Theta})q_p(\boldsymbol{\Theta}^*, \mathbf{D}^* \mid \boldsymbol{\Theta}, \mathbf{D})}\right\} \\ &= \min\left\{1, \frac{f(\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \mid \boldsymbol{\Theta}^*, \mathbf{D}^*)\mathbb{1}\left[\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon\right]\mathcal{M}(\mathbf{D}^* \mid \boldsymbol{\Theta}^*)f(\boldsymbol{\Theta}^*)q_p(\boldsymbol{\Theta}, \mathbf{D} \mid \boldsymbol{\Theta}^*, \mathbf{D}^*)}{f(\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \mid \boldsymbol{\Theta}, \mathbf{D})\mathbb{1}\left[\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon\right]\mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta})f(\boldsymbol{\Theta})q_p(\boldsymbol{\Theta}^*, \mathbf{D}^* \mid \boldsymbol{\Theta}, \mathbf{D})}\right\} \end{aligned} \tag{4.9}$$

where $q_p(\boldsymbol{\Theta}^*, \mathbf{D}^* \mid \boldsymbol{\Theta}, \mathbf{D})$ is the proposal distribution. ABC-MCMC proposes a new move by breaking up the proposal step into four conditionally independent steps, these are:

1. Propose a move from $\boldsymbol{\Theta}$ to $\boldsymbol{\Theta}^*$ using a proposal distribution $q(\boldsymbol{\Theta}^* \mid \boldsymbol{\Theta})$.
2. Simulate a data set $\mathbf{D}^*$ from the forward model $\mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}^*)$.
3. Calculate the distance function $\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}})$.
4. Proceed if $\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon$; otherwise reject $\boldsymbol{\Theta}^*$, and stay at $\boldsymbol{\Theta}$.

These steps divide the proposal function $q_p(\boldsymbol{\Theta}^*, \mathbf{D}^* \mid \boldsymbol{\Theta}, \mathbf{D})$ into four independent (conditionally) components, i.e.,

$$q_p(\boldsymbol{\Theta}^*, \mathbf{D}^* \mid \boldsymbol{\Theta}, \mathbf{D}) = q(\boldsymbol{\Theta}^* \mid \boldsymbol{\Theta}) \mathcal{M}(\mathbf{D}^* \mid \boldsymbol{\Theta}^*) f(\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \mid \boldsymbol{\Theta}^*, \mathbf{D}^*) \mathbb{1}\Big[\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon\Big] \tag{4.10}$$

Now, if we substitute Equation 4.10 into Equation 4.9, we get

$$\mathcal{A}(\{\boldsymbol{\Theta}, \mathbf{D}\}, \{\boldsymbol{\Theta}^*, \mathbf{D}^*\}) = \min\left\{1, \frac{f(\boldsymbol{\Theta}^*) q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^*)}{f(\boldsymbol{\Theta}) q(\boldsymbol{\Theta}^* \mid \boldsymbol{\Theta})}\right\} = \mathcal{A}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^*) \tag{4.11}$$

This proves that ABC-MCMC is a special case of the MH sampler, where the target distribution is the ABC posterior $f_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$. For a derivation of the ABC-MCMC sampler using the detailed balance equation, the reader may refer to the reference [83]. Note that if the proposal distribution is symmetric, i.e., $q(\boldsymbol{\Theta}^* \mid \boldsymbol{\Theta}) = q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^*)$, the acceptance probability depends only on the prior distribution as $\mathcal{A}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^*) = \min\Big\{1, \frac{f(\boldsymbol{\Theta}^*)}{f(\boldsymbol{\Theta})}\Big\}$. Additionally, if the prior is uniform, the acceptance probability turns into unity, which implies that all the proposed samples of $\boldsymbol{\Theta}$ will be accepted if they satisfy the accept-reject condition $\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon$.

A possible demerit of the ABC-MCMC scheme is that a very small tolerance threshold can cause the Markov chain to get stuck in low probability regions for a very long time, resulting in longer burn-in, thus longer computational time. To avoid this, one can initialize the algorithm by generating $\boldsymbol{\Theta}^{(1)}$ directly from $f_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$ using the ABC-RS algorithm (Algorithm 6).

Example

The computational advantage of the ABC-MCMC algorithm can be verified by solving the same problem that was solved using ABC-RS in the previous section. The same objective of inferring the correlation coefficient ρ_c can be achieved by choosing the following parameters of the algorithm:

1. Prior distribution: $f(\rho_c) = \mathcal{U}[-1, 1]$.
2. Forward model: $\mathcal{M}(\mathbf{D} \mid \rho_c) = f(x_1, x_2) = \dfrac{1}{2\pi\sqrt{1-\rho_c^2}} \exp\left[-\dfrac{x_1^2 - 2\rho_c x_1 x_2 + x_2^2}{2(1-\rho_c^2)}\right]$
3. Proposal distribution: $q(\rho_c^* \mid \rho_c) = \mathcal{N}(\rho_c, 0.2^2)$.
4. Distance function: $\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) = \left|r(\mathbf{D}) - r(\mathbf{D}_{\text{obs}})\right|$.
5. Tolerance threshold: $\epsilon = 0.005$.

Note that these are the same parameters used in previous examples in Section 3.5 and Section 4.3.1. To initialize the algorithm, ABC-RS is used to draw the initial sample directly from the ABC posterior $f_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$. The algorithm generated two Markov chains of lengths equal to 1×10^5 samples. Since the algorithm is initialized with a sample from the target ABC posterior itself, there is no need to discard the initial samples for burn-in. The estimation results are plotted in Figure 4.3, which shows that both Markov chains quickly converged to the high probability target region. Both

ABC posteriors estimated through ABC-MCMC can be seen to match well with the target posterior.

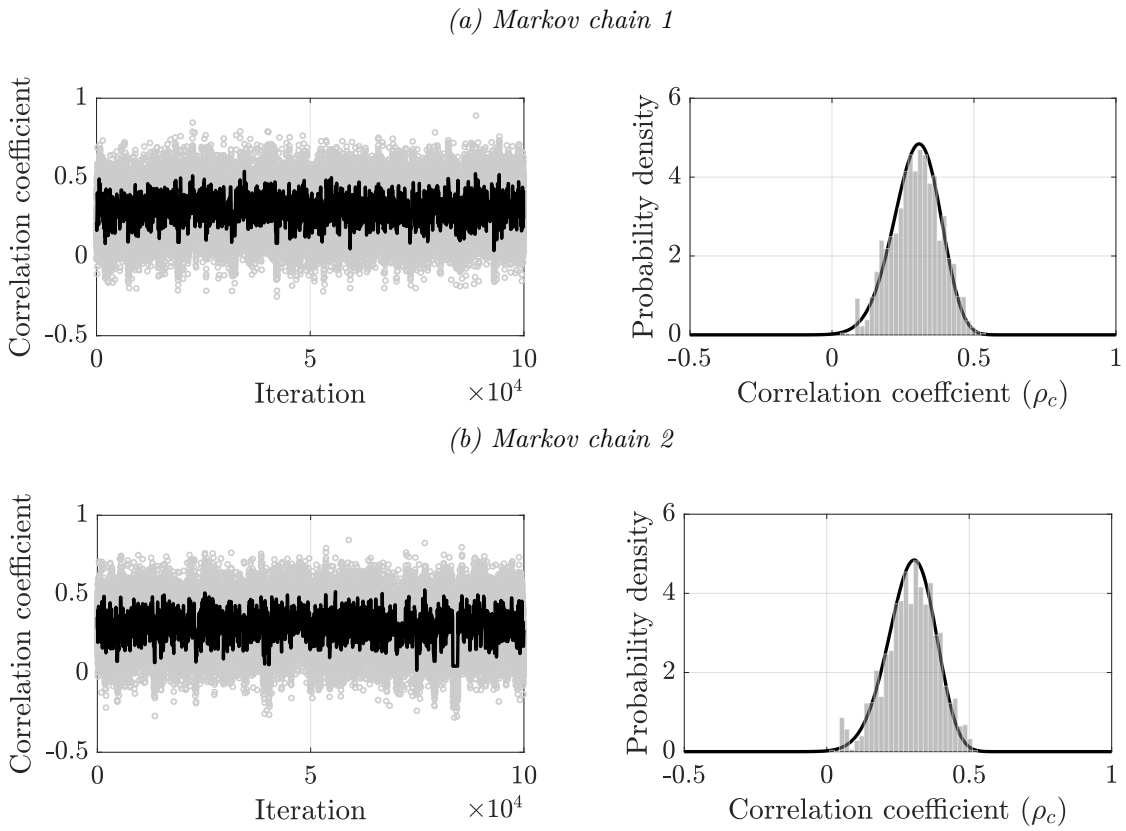


Figure 4.3: The Markov chains and the corresponding histograms of posterior samples generated from two independent runs of the ABC-MCMC sampler.

4.3.3 ABC using Subset Simulation

The likelihood-free MCMC method inherently suffers from the long burn-in issue in cases where diffuse priors are used or the parameter space is very high-dimensional. This issue is suitably handled by the sequential ABC samplers. These samplers propagate a set of parameter samples through a sequence of intermediate tolerance thresholds until they reach the target distribution. Several well-known sequential samplers, such as partial rejection control [119, 120], sequential importance sampling [39, 43, 129], population Monte Carlo [13], and subset simulation [30] have been combined within

the ABC framework. Using various benchmark examples, Chiachio et al. [30] showed that the subset simulation based ABC algorithm outperforms other variants of sequential ABC in terms of computational efficiency.

To avoid complexities related to "curse-of-dimensionality" problems, Chiachio et al. [30] proposed the efficient approximate Bayesian computation using subset simulation (ABC-SS) scheme. ABC-SS integrates the efficient rare event simulator subset simulation [7] with ABC as a sampling scheme. The subset simulation method was originally proposed by Au and Beck [7] to calculate a small failure probability (a rare event) by transforming it into a product of a sequence of larger conditional probabilities. This target is fulfilled by partitioning a failure domain into a nested sequence of failure subdomains. Similarly, by using subset simulation within the ABC framework, ABC-SS generates conditional posterior samples from a nested sequence of subdomains defined by the ABC distance function $\rho(\mathbf{D}, \mathbf{D}_{\text{obs}})$ and a set of tolerance thresholds: $\epsilon^{(1)} \geq \epsilon^{(2)} \geq \cdots \geq \epsilon^{(S)}$, where S is the total number of subdomains or simulation levels. The advantage of the ABC-SS algorithm is that it is not necessary to choose these tolerances in advance since the threshold levels can be chosen in an adaptive manner within the algorithm. The steps to implement ABC-SS are as follows:

1. First, generate n_0 number of samples of $\{\boldsymbol{\Theta}^{(k)}, \mathbf{D}^{(k)}\} \sim \mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}) f(\boldsymbol{\Theta})$, and for each case, evaluate $\rho^{(k)} = \rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}})$; n_0 is the number of model simulations at each simulation level. Then, sort and renumber the samples $\{\boldsymbol{\Theta}^{(k)}, \rho^{(k)}\}$ according to an ascending order of distance values: $\rho^{(1)} \leq \rho^{(2)} \leq \cdots \leq \rho^{(n_0)}$.

2. Select an acceptance probability p_0 such that $n_0 p_0$ and $1/p_0$ are integers. Set the first tolerance level $\epsilon^{(1)} = \rho^{(n_0 p_0)}$. Keep the first $n_0 p_0$ samples of $\{\boldsymbol{\Theta}^{(k)}, \rho^{(k)}\}$ and discard the rest of the samples. These $n_0 p_0$ samples of $\boldsymbol{\Theta}^{(k)}$ are assumed to be drawn from the ABC posterior $f_{\epsilon^{(1)}}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$ with acceptance probability p_0.

3. To generate more samples from the target density $f_{\epsilon^{(1)}}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$, the MCMC method is implemented. An MCMC sampler choose each $\{\boldsymbol{\Theta}^{(k)}, \rho^{(k)}\}$, $k = 1, 2, \cdots, n_0 p_0$, as a seed sample and generate $(1/p_0 - 1)$ new samples, making a total of $\{n_0 p_0 (1/p_0 - 1) + n_0 p_0\} = n_0$ samples. Once again, sort and renumber the samples $\{\boldsymbol{\Theta}^{(k)}, \rho^{(k)}\}$ according to an ascending order of new distance values: $\rho^{(1)} \leq \rho^{(2)} \leq \cdots \leq \rho^{(n_0)}$. Au and Beck [7] proposed an MCMC sampler called modified Metropolis (MM) algorithm (see Appendix C) that can be used for sampling while using subset simulation.

4. From the newly generated n_0 samples of $\{\boldsymbol{\Theta}^{(k)}, \rho^{(k)}\}$ in the previous step, keep the first $n_0 p_0$ samples of $\{\boldsymbol{\Theta}^{(k)}, \rho^{(k)}\}$ and discard the rest of the samples. Set the next tolerance level as $\epsilon^{(2)} = \rho^{(n_0 p_0)}$. These $n_0 p_0$ samples of $\boldsymbol{\Theta}^{(k)}$ are assumed to follow the ABC posterior $f_{\epsilon^{(2)}}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$. Since the new posterior samples are a result of conditional sampling, their acceptance probability will be p_0^2.

5. Repeat steps (3) and (4) to sequentially generate conditional posterior samples from $f_{\epsilon^{(s)}}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$, $s = 3, 4, \cdots, S$. The final samples of $\boldsymbol{\Theta}^{(k)}$ are the posterior samples drawn from the target ABC posterior $f_{\epsilon^{(S)}}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$ with an acceptance probability of p_0^S.

The pseudocode of the ABC-SS sampler is provided in Algorithm 8 and the pseudocode of the MM sampler, which can be directly integrated within Algorithm 8, is provided in Algorithm Algorithm 14. More detailed discussion on the properties and the computational convenience of the ABC-SS algorithm is beyond the scope of this thesis; interested readers may refer to reference [30].

Algorithm 8 ABC-SS sampler

for $s = 1$ to S **do**
  if $s = 1$ **then**
    for $k = 1$ to n_0 **do**
      Generate $\boldsymbol{\Theta}^{(s,k)} \sim f(\boldsymbol{\Theta})$
      Simulate $\mathbf{D}^* \sim \mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}^{(s,k)})$
      Evaluate $\rho^{(s,k)} = \rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}})$
    end for
    Renumber k index of $\{\boldsymbol{\Theta}^{(s,k)}, \rho^{(s,k)}\}$ by sorting $\rho^{(s,1)} \leq \rho^{(s,2)} \leq \cdots \leq \rho^{(s,n_0)}$
    Set $\epsilon^{(s)} = \rho^{(s,n_0 p_0)}$
  else
    for $l = 1$ to $n_0 p_0$ **do**
      Select a seed sample $\boldsymbol{\Theta}^{(s,h+1)} = \boldsymbol{\Theta}^{(s-1,l)}$, where $h = (1/p_0)(l-1)$
      for $r = 2$ to $1/p_0$ **do**
        Use MM to generate $\{\boldsymbol{\Theta}^{(s,h+r)}, \rho^{(s,h+r)}\}$ from $\{\boldsymbol{\Theta}^{(s,h+r-1)}, \rho^{(s,h+r-1)}\}$
      end for
    end for
    Renumber k index of $\{\boldsymbol{\Theta}^{(s,k)}, \rho^{(s,k)}\}$ by sorting $\rho^{(s,1)} \leq \rho^{(s,2)} \leq \cdots \leq \rho^{(s,n_0)}$
    Set $\epsilon^{(s)} = \rho^{(s,n_0 p_0)}$
  end if
end for

4.4 Prior Selection for ABC

In Bayesian inference, the prior distribution works as a key element to represent the prior belief about an uncertain model parameter. According to the Bayesian interpretation of probability, the prior belief is obtained from an individual's past experience and personal judgment. Then, based on the newly available information/data, the prior belief/distribution is updated to the posterior belief/distribution of the model parameter using the Bayes' theorem. The main difference between the likelihood-based and the likelihood-free approach is that the former uses the probability distribution of newly available data for updating, whereas the latter uses the data to reject specific parameter samples that generate data far away from the observed data (the closeness is measured

using the ABC distance function). For both approaches, a prior distribution is always available since it is selected by an individual; however, a reasonable prior choice and large sample size will have minimum effects on the inferences made by the posterior samples. In contrast, if the sample size is small or the available information is indirect in nature (involves noisy measurements or hidden states), then the selection of a prior distribution becomes crucial. In practice, the dependence of the posterior samples on prior specification can be checked by a sensitivity analysis, i.e., by comparing the posterior inferences or model predictions using different prior choices that are reasonable [49]. Minimal effect on posterior inference can be also be obtained by selecting uninformative diffused priors when no prior information or expert judgment is available for a set of model parameters. However, diffused priors may put computational burdens on the basic rejection and MCMC-based ABC samplers due to their large variances resulting in larger domains of parameter choices; the sequential ABC samplers in such situations may stand out and perform better. Lastly, the prior selection for ABC must be motivated by the fact that sampling from a prior is straightforward and easy, so that the sampler does not get stuck or spend more time on generating prior samples that might render ABC inefficient.

4.5 ABC Distance Function

The distance function in the likelihood-free Bayesian inference framework is equivalent to the likelihood function in the likelihood-based framework. Although it plays a key role in the process of parameter inference, there is no theoretical justification or practical guideline available in the literature that describes what distance function should be chosen for what type of data [142]. For instance, Chiachio et al. [30] applied the ABC-SS algorithm for parameter inference of a moving average process of order two

using the quadratic distance between the two first autocovariances of the simulated and observed data; whereas, in an another example, the authors used the Euclidean (ℓ_2) distance as a metric for parameter inference of a single degree of freedom (DOF) linear oscillator that is subjected to white noise excitation. Vakilzadeh et al. [135] studied the performance of ABC-SS on hierarchical state-space models. The authors considered two DOF linear and nonlinear structures where they used the Chebyshev distance (or infinity norm) as the ABC distance function. In an another study, Vakilzadeh et al. [136] used the Euclidean norm to calculate the distance function between the simulated and observed measurements, where they focused on dynamic state-space models such as single DOF bilinear oscillator and three DOF nonlinear structure. Abdessalem et al. [3, 4] studied nonlinear system identification problems using different ABC distance functions such as normalized mean square error between the observed and simulated displacements, Euclidean distance of summaries of displacement data, and Euclidean distance between the simulated and observed PDFs of the acceleration. Chen et al. [27] presented ABC as an efficient tool to analyze time-censored lifetime data where the authors preferred to use the Manhattan (ℓ_1) distance between the observed and simulated number of replacements. Hazra et al. [60, 61] implemented ABC to analyze degradation measurements using the Euclidean distance between observed and simulated degradation data. The Euclidean distance is clearly the most commonly used distance function in the literature, although Chebyshev and Manhattan distances have been used a number of times. In summary, it can be noted that the ABC distance function does not follow any particular form, and it can be formulated according to the problem at hand and the quantity of interest that needs to be estimated.

4.6 ABC for Noisy Degradation Data

To avoid evaluating computationally expensive likelihoods that are the result of noise and detection errors in the degradation data, one may choose to employ the likelihood-free ABC schemes. However, the question remains, how do we handle imperfect data in the ABC framework? Generally, ABC uses the underlying degradation model to simulate true or perfect data. Then, in case of flaw growths, the data can be contaminated by adding noise simulated from its assumed model or probability distribution. However, simulating imperfect data in case of flaw generation is slightly difficult since one needs to mimic the process of flaw detection using a POD function along with the fact that the measured flaw growths are then made noisy by adding simulated noise. Finally, the simulated imperfect data can be compared with the observed imperfect data to estimate the ABC posterior.

4.7 Performance Assessment of ABC Samplers

The performance of any ABC sampler is mainly dependent on the selection of the tolerance threshold, and to achieve higher accuracy, a smaller tolerance threshold is always preferred. However, if a very small tolerance is selected, the sampler may keep rejecting the parameter samples for an indefinite period and may get trapped into a low probability region. Thus, to maintain a balance between the accuracy and the computational cost, the tolerance needs to be selected in a way so that the parameter estimates are reasonably accurate for quantities that are inferred using the posterior samples. To achieve that goal, first, the posterior density estimates (joint and marginals) are compared with the results obtained using the likelihood-based MCMC sampler; apparently, the results of the likelihood-based MCMC scheme are assumed gold standard. Then, the model predictions from both samplers can be compared to

assess the performance of the ABC samplers. Note that this process can only be applied to problems (e.g., simulation problems) that can be solved using both likelihood-based and likelihood-free approaches, to select a suitable tolerance threshold and other ABC hyperparameters. Once the ABC hyperparameters are selected, the sampling process can be performed for models that involve more complex likelihood functions.

4.8 Limitations of ABC

The ABC method comes with a few limitations due to the assumptions and approximations made to achieve higher efficiency with reasonable accuracy. The primary limitation of ABC is that it depends on the forward simulation of a model that is assumed to best describe the data at hand. Thus, if any model with a complex likelihood can not be directly used as a data simulator, then ABC simply don't turn out to be a reasonable choice. An example of such a model is the Hougaard process model [64]. Since there is no closed-form expression available for its density function, forward simulation of such a model is not straightforward.

ABC may generate biased results due to the tolerance not being zero. Also, the distance function plays a key role; different distance measures may generate different results. The fact that ABC generates posterior samples from $f(\boldsymbol{\Theta} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon)$ instead of the true posterior $f(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$ makes it difficult for an individual to choose the right tolerance value and a suitable distance function in practice. More investigations are needed to understand, untangle, and quantify the errors introduced by these approximations so that better selections can be made, which is an active area of research in the current times.

In the case of a high dimensional data set, low dimensional summary statistics are preferred. This helps ABC to achieve faster acceptance. However, finding suffi-

cient statistics is not always feasible for the problems where ABC is applicable, and poorly chosen summary statistics often lead to overestimation of the posterior parameter ranges. Moreover, as one increases the number of summary statistics, the performance of ABC seems to decrease drastically. Thus, only relevant summary statistics should be kept for formulating the ABC distance function so that the target quantity of interest can be inferred with reasonable accuracy. For a practical guidance on the selection of the summary statistics, the reader may refer to [18].

4.9 ABC using Hamiltonian Monte Carlo

This section presents a new ABC method that integrates the Hamiltonian Monte Carlo (HMC) method as its sampling scheme. Governed by Hamiltonian dynamics, the HMC method suppresses the random-walk behavior of the standard MCMC method and explore the target probability space more effectively and consistently. As a result, the proposed ABC-HMC method shows better mixing of samples in a Markov chain than the MCMC based ABC method. Meeds et al. [87] were the first to propose an ABC algorithm based on the HMC method where the authors considered a pseudo-marginal approach that consists of an approximate likelihood model (Gaussian) for the simulator $\mathcal{M}(\mathbf{D} \mid \mathbf{\Theta})$. On the other hand, this study proposes an approach based on a marginal sampler, that means the proposed ABC-HMC method completely avoids the use of any form of likelihood. To understand how HMC applies to ABC, one needs to know the basic idea of the HMC algorithm which is presented in the following subsection.

4.9.1 Hamiltonian Monte Carlo Sampler

The Hamiltonian Monte Carlo is a sampling scheme that utilizes the Hamiltonian dynamics to propose new moves and traverse through the probability space. Let us

consider an auxiliary variable $\mathbf{p}$ that is independent of the parameter vector $\boldsymbol{\Theta}$. The main goal of HMC is to draw samples from the target joint distribution $f(\boldsymbol{\Theta}, \mathbf{p}) = f(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) f(\mathbf{p})$, where $f(\mathbf{p})$ is commonly chosen to be a zero mean multivariate normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{M})$, where $\mathbf{M}$ is the covariance matrix.

In the context of Hamiltonian dynamics, $\boldsymbol{\Theta}$ is the position variable, $\mathbf{p}$ is the momentum variable, and $\mathbf{M}$ is the mass matrix. The equations of motion (for conservative systems) governing the time evolution of the variable pair $(\boldsymbol{\Theta}, \mathbf{p})$ can be written as

$$\frac{d\boldsymbol{\Theta}}{dt} = \frac{\partial H}{\partial \mathbf{p}}, \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \boldsymbol{\Theta}} \tag{4.12}$$

where $H(\boldsymbol{\Theta}, \mathbf{p})$ is the Hamiltonian that represents the total energy of a dynamical system. $H(\boldsymbol{\Theta}, \mathbf{p})$ is independent of time and can be represented as a sum of the potential energy $U(\boldsymbol{\Theta})$ and the kinetic energy $K(\mathbf{p})$, i.e., $H(\boldsymbol{\Theta}, \mathbf{p}) = U(\boldsymbol{\Theta}) + K(\mathbf{p})$. To connect the Hamiltonian with Bayesian inference, the potential and kinetic energies are typically defined as

$$U(\boldsymbol{\Theta}) = -\log \mathcal{L}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) - \log f(\boldsymbol{\Theta}), \quad K(\mathbf{p}) = \mathbf{p}^\top \mathbf{M}^{-1} \mathbf{p}/2 \tag{4.13}$$

By substituting the expression for the Hamiltonian $H(\boldsymbol{\Theta}, \mathbf{p})$ into Equation 4.12, we get

$$\frac{d\boldsymbol{\Theta}}{dt} = \mathbf{M}^{-1}\mathbf{p}, \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial U}{\partial \boldsymbol{\Theta}} \tag{4.14}$$

Accordingly, the joint distribution of $(\boldsymbol{\Theta}, \mathbf{p})$ can be written as

$$\begin{aligned} f(\boldsymbol{\Theta}, \mathbf{p}) = \frac{1}{Z} \exp\{-H(\boldsymbol{\Theta}, \mathbf{p})\} &= \frac{1}{Z} \exp\{-U(\boldsymbol{\Theta}) - K(\mathbf{p})\} \\ &= \frac{1}{Z} \mathcal{L}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) f(\boldsymbol{\Theta}) \exp\left\{ -\frac{\mathbf{p}^\top \mathbf{M}^{-1} \mathbf{p}}{2} \right\} \end{aligned} \tag{4.15}$$

where Z is a normalizing constant.

Samples from the target posterior distribution can be obtained by generating a Markov chain with the stationary distribution given in Equation 4.15, and marginalizing it afterwards. Thus, the HMC sampling method consists of two main steps: first, assuming $\boldsymbol{\Theta}'$ to be the current position variable, draw a new momentum variable $\mathbf{p}'$ from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{M})$; then, from the current state $(\boldsymbol{\Theta}', \mathbf{p}')$, propose a new state $(\boldsymbol{\Theta}^*, \mathbf{p}^*)$ by integrating the equations of motion (Equation 4.14) at a specific time. The integration is generally performed using a numerical integration scheme such as the commonly used leapfrog method [95]. To solve Equation 4.14, the leapfrog method works as follows:

$$\mathbf{p}(t + h/2) = \mathbf{p}(t) - (h/2)\frac{\partial U}{\partial \boldsymbol{\Theta}}(\boldsymbol{\Theta}(t)) \tag{4.16a}$$

$$\boldsymbol{\Theta}(t + h) = \boldsymbol{\Theta}(t) + h\mathbf{M}^{-1}\mathbf{p}(t + h/2) \tag{4.16b}$$

$$\mathbf{p}(t + h) = \mathbf{p}(t + h/2) - (h/2)\frac{\partial U}{\partial \boldsymbol{\Theta}}(\boldsymbol{\Theta}(t + h)) \tag{4.16c}$$

where h is an incremental time step; the process is repeated for L time steps. In case there is no direct availability of the gradient $\partial U/\partial \boldsymbol{\Theta}$, one may take help from standard numerical differentiation schemes [143]. The numerical integration scheme presented above may compromise the invariance property of the Hamiltonian that subsequently guarantees the inavriance of the target posterior density. To preserve this property, one needs to combine the Metropolis accept-reject rule, $\mathcal{A}(\boldsymbol{\Theta}', \boldsymbol{\Theta}^*) = \min\left\{1, \exp\left(H(\boldsymbol{\Theta}', p') - H(\boldsymbol{\Theta}^*, p^*)\right)\right\}$, on a newly proposed state. A pseudocode implementation of the HMC method is presented in Algorithm 9.

The mass matrix $\mathbf{M}$ should be symmetric and positive-definite; $\mathbf{M}$ is commonly assumed to be a scalar multiple of the identity matrix [143]. The incremental time step h should be sufficiently small so that the Hamiltonian is approximately preserved. On the other hand, the number of leapfrog steps L should be chosen in a way so

Algorithm 9 HMC sampler

1: Initialize $\boldsymbol{\Theta}^{(i)} \sim f(\boldsymbol{\Theta})$, set $i = 1$.
2: **for** $i = 1$ to $n - 1$ **do**
3: Select a seed sample $\boldsymbol{\Theta}' = \boldsymbol{\Theta}^{(i)}$ and draw $\mathbf{p}' \sim \mathcal{N}(\mathbf{0}, \mathbf{M})$.
4: Propose a move from $(\boldsymbol{\Theta}', \mathbf{p}')$ to $(\boldsymbol{\Theta}^*, \mathbf{p}^*)$ using L leapfrog steps.
5: Calculate $\mathcal{A}(\boldsymbol{\Theta}', \boldsymbol{\Theta}^*) = \min\left\{1, \exp\left(H(\boldsymbol{\Theta}', \mathbf{p}') - H(\boldsymbol{\Theta}^*, \mathbf{p}^*)\right)\right\}$.
6: Set $\boldsymbol{\Theta}^{(i+1)} = \boldsymbol{\Theta}^*$ with probability $\mathcal{A}(\boldsymbol{\Theta}', \boldsymbol{\Theta}^*)$; else, set $\boldsymbol{\Theta}^{(i+1)} = \boldsymbol{\Theta}'$.
7: **end for**

that the length of the trajectory hL is long enough to explore the points far from the current point. Algorithm 9 is applicable when the likelihood function $\mathcal{L}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$ is analytically or computationally tractable. The likelihood-free version of the algorithm is developed in the next subsection.

4.9.2 ABC-HMC Algorithm

The ABC-HMC method aims to draw samples from the ABC posterior $f_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}}) = f(\boldsymbol{\Theta} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon)$. To achieve this goal, the Hamiltonian needs to be modified according to the requirement of the ABC framework. The modified Hamiltonian can be formulated as

$$\begin{aligned} H(\boldsymbol{\Theta}, \mathbf{p}) &= -\log f(\boldsymbol{\Theta} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon) + \mathbf{p}^\top \mathbf{M}^{-1} \mathbf{p}/2 \\ &= -\log f(\boldsymbol{\Theta}) + \mathbf{p}^\top \mathbf{M}^{-1} \mathbf{p}/2 - \log \mathbb{1}[\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon] + \text{const.} \end{aligned} \tag{4.17}$$

Equation 4.17 is obtained by substituting

$$f(\boldsymbol{\Theta} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon) = \frac{f(\boldsymbol{\Theta}) \mathbb{1}[\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon]}{\int_{\boldsymbol{\Theta}} f(\boldsymbol{\Theta}) \mathbb{1}[\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon] d\boldsymbol{\Theta}} \tag{4.18}$$

where the integral at the denominator results in a constant term in Equation 4.17. This constant term can be ignored since it does not have any effect on the solution of the Hamilton's equations. Besides, the term $(-\log \mathbb{1}[\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon])$ in Equation 4.17

creates a potential barrier that guarantees that the Hamiltonian system does not reach the region outside of $[\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon]$ – a region with infinite potential energy [143]. The steps of the ABC-HMC algorithm are presented in Algorithm 10. It can be noticed

Algorithm 10 ABC-HMC sampler

1: Initialize $\boldsymbol{\Theta}^{(i)} \sim f(\boldsymbol{\Theta})$, set $i = 1$.
2: **for** $i = 1$ to $n - 1$ **do**
3: Select a seed sample $\boldsymbol{\Theta}' = \boldsymbol{\Theta}^{(i)}$ and draw $\mathbf{p}' \sim \mathcal{N}(\mathbf{0}, \mathbf{M})$.
4: Propose a move from $(\boldsymbol{\Theta}', \mathbf{p}')$ to $(\boldsymbol{\Theta}^*, \mathbf{p}^*)$ using L leapfrog steps.
5: Simulate $\mathbf{D}^* \sim \mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}^*)$.
6: Calculate $\mathcal{A}(\boldsymbol{\Theta}', \boldsymbol{\Theta}^*) = \min\Big\{1, \exp\Big(H(\boldsymbol{\Theta}', \mathbf{p}') - H(\boldsymbol{\Theta}^*, \mathbf{p}^*)\Big)\mathbb{1}[\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon]\Big\}$.
7: Set $\boldsymbol{\Theta}^{(i+1)} = \boldsymbol{\Theta}^*$ with probability $\mathcal{A}(\boldsymbol{\Theta}', \boldsymbol{\Theta}^*)$; else, set $\boldsymbol{\Theta}^{(i)} = \boldsymbol{\Theta}'$.
8: **end for**

that the likelihood-free HMC sampler has a modified acceptance probability that turns zero if the ABC distance function generates a value more than the tolerance threshold. In order to prove the convergence of the proposed algorithm, the detailed balance equations need to be satisfied. It is shown below that the stationary distribution of the chain is $f(\boldsymbol{\Theta}, \mathbf{p}, \mathbf{D} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon)$, which can be marginalized over $\mathbf{p}$ and $\mathbf{D}$ to produce the ABC posterior $f_\epsilon(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$.

Theorem. *$f(\boldsymbol{\Theta}, \mathbf{p}, \mathbf{D} \mid \rho(\mathbf{D}, \mathbf{D}_{obs}) \leq \epsilon)$ is the stationary distribution of the Monte Carlo Markov chain produced by ABC-HMC.*

Proof. Let $\mathcal{K}(\boldsymbol{\Theta}^*, \mathbf{p}^* \mid \boldsymbol{\Theta}', \mathbf{p}')$ be the transition mechanism and $q(\boldsymbol{\Theta}^*, \mathbf{p}^* \mid \boldsymbol{\Theta}', \mathbf{p}')$ be the proposal mechanism of the chain for transitioning from $(\boldsymbol{\Theta}', \mathbf{p}')$ to $(\boldsymbol{\Theta}^*, \mathbf{p}^*)$. Since transitioning from $(\boldsymbol{\Theta}', \mathbf{p}')$ to $(\boldsymbol{\Theta}^*, \mathbf{p}^*)$ is a deterministic event in Hamiltonian dynamics, the proposal function can take either a value of 0 or 1, i.e.,

$$q(\boldsymbol{\Theta}^*, \mathbf{p}^* \mid \boldsymbol{\Theta}', \mathbf{p}') = \begin{cases} 1, & \text{if transition occurs} \\ 0, & \text{otherwise} \end{cases} \tag{4.19}$$

Let us denote the acceptance probability as

$$\mathcal{A}\Big\{(\boldsymbol{\Theta}', \mathbf{p}'), (\boldsymbol{\Theta}^*, \mathbf{p}^*)\Big\} = \min\Big\{1, \exp\Big(H(\boldsymbol{\Theta}', \mathbf{p}') - H(\boldsymbol{\Theta}^*, \mathbf{p}^*)\Big)\mathbb{1}[\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon]\Big\} \tag{4.20}$$

Now, assuming the transition has occurred, if we select $\boldsymbol{\Theta}^*$ such that

$$\exp\Big(H(\boldsymbol{\Theta}', \mathbf{p}') - H(\boldsymbol{\Theta}^*, \mathbf{p}^*)\Big) \leq 1, \quad \boldsymbol{\Theta}^* \neq \boldsymbol{\Theta}' \tag{4.21}$$

Then, we can write

$$\begin{aligned}
&f(\boldsymbol{\Theta}', \mathbf{p}', \mathbf{D}' \mid \rho(\mathbf{D}', \mathbf{D}_{\text{obs}}) \leq \epsilon)\mathcal{K}(\boldsymbol{\Theta}^*, \mathbf{p}^* \mid \boldsymbol{\Theta}', \mathbf{p}') \\
&= f(\boldsymbol{\Theta}', \mathbf{p}', \mathbf{D}' \mid \rho(\mathbf{D}', \mathbf{D}_{\text{obs}}) \leq \epsilon)q(\boldsymbol{\Theta}^*, \mathbf{p}^* \mid \boldsymbol{\Theta}', \mathbf{p}')\mathcal{M}(\mathbf{D}^* \mid \boldsymbol{\Theta}^*)\mathcal{A}\Big\{(\boldsymbol{\Theta}', \mathbf{p}'), (\boldsymbol{\Theta}^*, \mathbf{p}^*)\Big\} \\
&= \frac{f(\boldsymbol{\Theta}', \mathbf{p}')\mathcal{M}(\mathbf{D}' \mid \boldsymbol{\Theta}')\mathbb{1}[\rho(\mathbf{D}', \mathbf{D}_{\text{obs}}) \leq \epsilon]}{\int_{\mathbf{D}}\int_{\mathbf{p}}\int_{\boldsymbol{\Theta}} f(\boldsymbol{\Theta}, \mathbf{p})\mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta})\mathbb{1}[\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon]d\boldsymbol{\Theta}d\mathbf{p}d\mathbf{D}}\Big\{q(\boldsymbol{\Theta}^*, \mathbf{p}^* \mid \boldsymbol{\Theta}', \mathbf{p}')\mathcal{M}(\mathbf{D}^* \mid \boldsymbol{\Theta}^*) \\
&\qquad \times \exp\Big(H(\boldsymbol{\Theta}', \mathbf{p}') - H(\boldsymbol{\Theta}^*, \mathbf{p}^*)\Big)\mathbb{1}[\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon]\Big\}
\end{aligned} \tag{4.22}$$

Substituting (from Equation 4.15)

$$f(\boldsymbol{\Theta}', \mathbf{p}')\exp\Big(H(\boldsymbol{\Theta}', \mathbf{p}') - H(\boldsymbol{\Theta}^*, \mathbf{p}^*)\Big) = f(\boldsymbol{\Theta}^*, \mathbf{p}^*), \quad \text{and} \tag{4.23}$$

$$q(\boldsymbol{\Theta}^*, \mathbf{p}^* \mid \boldsymbol{\Theta}', \mathbf{p}') = 1 = q(\boldsymbol{\Theta}', \mathbf{p}' \mid \boldsymbol{\Theta}^*, \mathbf{p}^*) \tag{4.24}$$

we get

$$\begin{aligned}
&= \frac{f(\boldsymbol{\Theta}^*, \mathbf{p}^*)\mathcal{M}(\mathbf{D}^* \mid \boldsymbol{\Theta}^*)\mathbb{1}[\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon]}{\int_{\mathbf{D}}\int_{\mathbf{p}}\int_{\boldsymbol{\Theta}} f(\boldsymbol{\Theta}, \mathbf{p})\mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta})\mathbb{1}[\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon]d\boldsymbol{\Theta}d\mathbf{p}d\mathbf{D}}\Big\{q(\boldsymbol{\Theta}', \mathbf{p}' \mid \boldsymbol{\Theta}^*, \mathbf{p}^*)\mathcal{M}(\mathbf{D}' \mid \boldsymbol{\Theta}') \\
&\qquad \times \mathbb{1}[\rho(\mathbf{D}', \mathbf{D}_{\text{obs}}) \leq \epsilon]\Big\} \\
&= f(\boldsymbol{\Theta}^*, \mathbf{p}^*, \mathbf{D}^* \mid \rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon)\mathcal{K}(\boldsymbol{\Theta}', \mathbf{p}' \mid \boldsymbol{\Theta}^*, \mathbf{p}^*)
\end{aligned} \tag{4.25}$$

The arguments for $q(\boldsymbol{\Theta}^*, \mathbf{p}^* \mid \boldsymbol{\Theta}', \mathbf{p}') = 0$ (i.e., non-transition) and for $\exp\Big(H(\boldsymbol{\Theta}', \mathbf{p}') - H(\boldsymbol{\Theta}^*, \mathbf{p}^*)\Big) > 1$ are analogous. Thus, the detailed balance equations are satisfied for $f(\boldsymbol{\Theta}, \mathbf{p}, \mathbf{D} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon)$. This proves that $f(\boldsymbol{\Theta}, \mathbf{p}, \mathbf{D} \mid \rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon)$ is the stationary distribution of the Monte Carlo Markov chain produced by ABC-HMC.

Example

This example compares the performances of the standard ABC-MCMC and the proposed ABC-HMC algorithms, particularly at the distribution tails. The example includes sampling from a target posterior density that has a sharp mode along with a very small mode next to it. A similar example can be found in reference [9]. Suppose an observation $D_{\text{obs}} = 0$ is generated from a Gaussian mixture model, $0.45\mathcal{N}(\theta, 1) + 0.45\mathcal{N}(\theta, 1/10^2) + 0.1\mathcal{N}(\theta - 5, 1)$. Assuming a normal prior $f(\theta) = \mathcal{N}(0, 5^2)$ for the parameter θ, the posterior distribution can be written as

$$f(\theta \mid D_{\text{obs}} = 0) \propto \Big(0.45\mathcal{N}(\theta, 1) + 0.45\mathcal{N}(\theta, 1/10^2) + 0.1\mathcal{N}(\theta - 5, 1)\Big) \times \mathcal{N}(0, 5^2) \quad (4.26)$$

To implement the ABC algorithms, we select a distance function $\rho(D, D_{\text{obs}}) = |D - D_{\text{obs}}|$ and a tolerance threshold $\epsilon = 0.02$. The algorithms are initialized from the sample $\theta = 0$ and run for 5×10^6 iterations.

One common problem in any MCMC method is that the sampler gets trapped either in a high or a low probability region due to their inability to jump from one region to another – resulting in a poor mixing of the chain. The issue can be resolved by tuning the parameters of the proposal distribution. In ABC-MCMC, we select a Gaussian proposal distribution $q(\theta \mid \theta') = \mathcal{N}(\theta', \sigma^2)$, where θ' is the current sample and σ is the proposal standard deviation – the parameter which needs tuning. In ABC-HMC, the proposal is a deterministic function governed by the Hamiltonian dynamics, where the time step parameter h and leapfrog step parameter L need tuning.

Figure 4.4 shows the ABC-MCMC posteriors that are estimated using different σ values ranging from 0.25 to 1 in steps of 0.25. Compared to the true posterior distribution, the ABC-MCMC posteriors with $\sigma < 0.75$ can be seen struggling with exploring the small mode around 5. However, it appears that the sampler can successfully visit

the small mode when the parameter $\sigma \geq 0.75$. Thus, for further analysis, a conservative choice of $\sigma = 1.0$ is made for the proposal standard deviation.

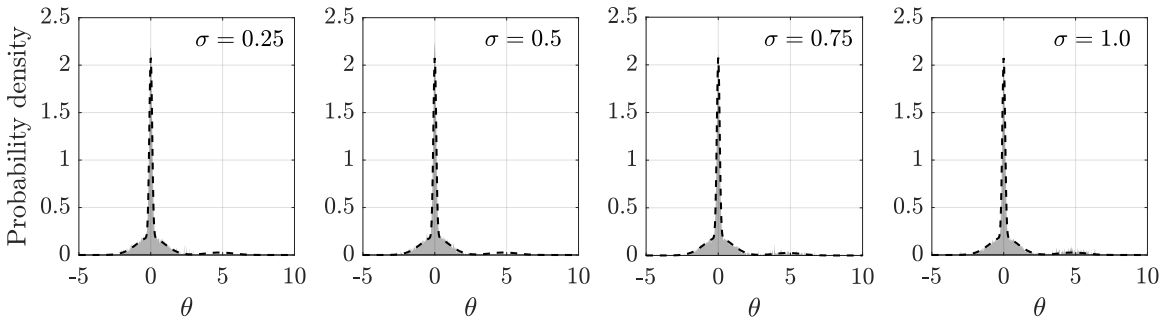


Figure 4.4: Posterior distributions of θ obtained using ABC-MCMC with different σ values. The black broken line represents the true posterior distribution, and the gray area represents the ABC-MCMC posterior.

Similar numerical experiments can be performed to tune the parameters of the ABC-HMC algorithm. Figure 4.5 shows the ABC-HMC posteriors estimated using a fixed value of $L = 20$ and different values of h ranging from 0.025 to 0.1 in steps of 0.025. It can be seen that the performance of ABC-HMC with $h < 0.075$ is the worst at the distribution tails. However, $h \geq 0.075$ shows better performance, which means the ABC-HMC sampler does not get trapped in the high probability region. For further analysis, a suitable pair of the parameters can be chosen as $h = 0.1$ and $L = 20$.

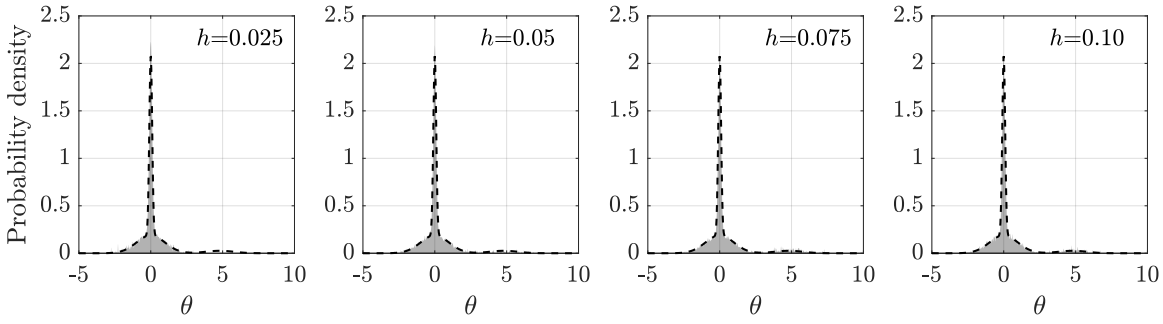


Figure 4.5: Posterior distributions of θ obtained using ABC-HMC with different h values. The black broken line represents the true posterior distribution, and the gray area represents the ABC-HMC posterior.

The algorithms are implemented with the selected parameters as discussed above. The sample iterations of both algorithms are shown in Figure 4.6. Notice that ABC-

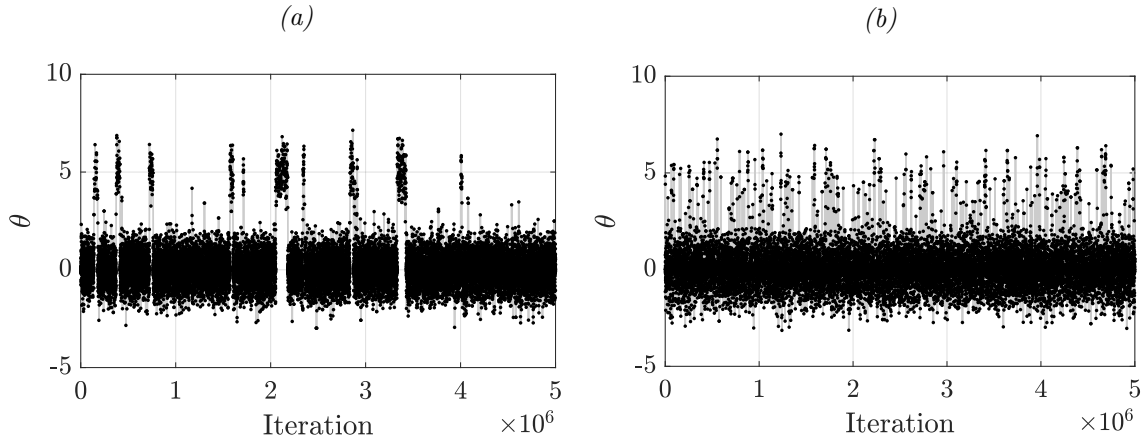


Figure 4.6: Sample iterations of θ obtained using (a) ABC-MCMC and (b) ABC-HMC.

HMC shows better mixing of the chain by not getting trapped in either of the modes. Whereas, ABC-MCMC struggles to jump between the two modes of the target posterior distribution resulting in poor mixing of the chain. The performances of both the algorithms in chain mixing can be quantitatively studied using the sample ACF plot presented in Figure 4.7. The figure suggests that ABC-MCMC needs a thinning

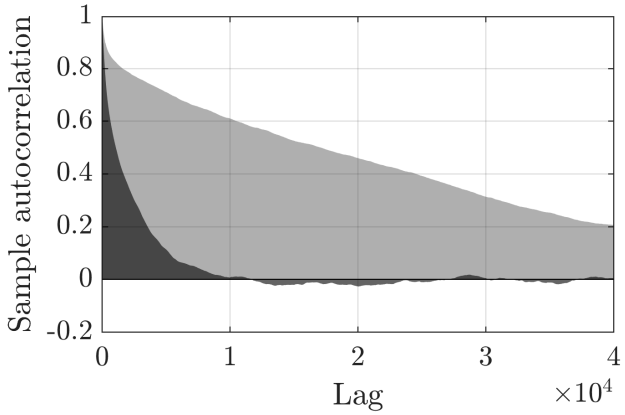


Figure 4.7: Autocorrelations (normalized) within the θ samples obtained from ABC-MCMC (light gray) and ABC-HMC (deep gray).

interval of around 40,000 to effectively generate uncorrelated samples. As a result, a large number of samples need to be discarded. On the contrary, for ABC-HMC, the figure suggests a thinning interval of only around 5000 to generate similar uncorrelated samples. This proves that the performance of the proposed ABC-HMC algorithm is better than the standard ABC-MCMC method in providing good mixing of the samples and exploring the tails of target posterior distributions.

4.9.3 ABC-HMC with Subset Simulation

Although the ABC-HMC algorithm shows better mixing properties than the conventional ABC-MCMC method, it may struggle to converge when single or multi-dimensional problems are dealt with uninformative or diffuse priors. To avoid this problem, we propose to integrate ABC-HMC with subset simulation. As mentioned before in Subsection 4.3.3, subset simulation enables ABC to perform in a similar way by defining a decreasing sequence of tolerance thresholds, i.e., $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_S$, where S is the total number of simulation levels, and ϵ_S is the target tolerance threshold [30]. In fact, one can select the the sequence of tolerances adaptively as described below.

Recall that ABC-SS starts by generating n_0 number of samples of $\{\boldsymbol{\Theta}, \mathbf{D}\}$, i.e., $\{\boldsymbol{\Theta}^{(k)}, \mathbf{D}^{(k)}\}_{k=1}^{n_0} \sim \mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}) f(\boldsymbol{\Theta})$. The distance function is then evaluated pointwise, i.e., $\rho^{(k)} = \rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}})$, for all the simulated data sets, and the values are sorted and renumbered according to the following order: $\rho^{(1)} < \rho^{(2)} < \cdots < \rho^{(n_0)}$. Next, the samples $\{\boldsymbol{\Theta}^{(k)}\}_{k=1}^{n_0}$ are renumbered according to the order of the distance values. Finally, a probability p_0 is selected (such that $n_0 p_0$ and $1/p_0$ are integers), and the first $n_0 p_0$ samples of $\boldsymbol{\Theta}$ are kept for further analysis; the rest of the samples are discarded. This step makes $(n_0 p_0)$th distance value equal to the tolerance threshold of the first simulation level, i.e., $\epsilon_1 = \rho^{(n_0 p_0)}$, and the samples $\{\boldsymbol{\Theta}^{(k)}\}_{k=1}^{n_0 p_0}$ belong to the ABC posterior $f_{\epsilon_1}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$. This completes the first simulation level. At the

second simulation level, $\{\boldsymbol{\Theta}^{(k)}\}_{k=1}^{n_0 p_0}$ samples from the previous level are considered as seed samples and from each seed sample, $(1/p_0 - 1)$ new samples are generated using an MCMC scheme with a fixed tolerance ϵ_1. Similar to the first simulation level, the sorting and renumbering of the distance values and parameter samples are carried out to find the tolerance of the second level ϵ_2 and the seed samples for the next level. The sampling process is repeated for each simulation level until the algorithm reaches the very last stage that is the simulation level S (see Algorithm 8).

The MCMC sampling at the intermediate levels of ABC-SS are traditionally performed using the MM algorithm in Algorithm 14. The MM algorithm identifies the independent components of the parameter vector, i.e., $\boldsymbol{\Theta} = \{\Theta_1, \Theta_2, \cdots, \Theta_C\}$, and update the components $\{\Theta_i\}_{i=1}^{C}$ individually using independent priors $\{f_i(\Theta_i)\}_{i=1}^{C}$ and independent proposals $\{q_i(\Theta_i \mid \Theta_i')\}_{i=1}^{C}$; where $\{\Theta_i')\}_{i=1}^{C}$ are the current positions of the parameter components. The idea of component-wise update in MM sampler can also be incorporated into ABC-HMC to improve its mixing further. Following the similar idea, we propose a modified version of the HMC algorithm to be integrated with ABC-SS. Similar to the MM sampler, the components of the parameter vector $\{\Theta_i\}_{i=1}^{C}$ are individually updated using independent priors $\{f_i(\Theta_i)\}_{i=1}^{C}$ and individual momentum variables $p_i \sim \mathcal{N}(0, M_i)$, $i = 1, 2, \cdots, C$. Furthermore, the individual Hamiltonian $H_i(\Theta_i, p_i) = -\log f_i(\Theta_i) + p_i^{\mathrm{T}} M_i^{-1} p_i/2$ can be used to propose the independent parameter components. The modified HMC (MHMC) algorithm for ABC-SS is presented in Algorithm 11. In this study, the conventional ABC-SS algorithm integrated with the MM sampler is denoted as ABC-SS(MM), and similarly, ABC-SS integrated with the proposed MHMC sampler is denoted as ABC-SS(MHMC). The performances of both algorithms in terms of sample mixing and parameter space exploration are compared in the following example.

Algorithm 11 MHMC sampler for ABC-SS

1: **for** $i = 1$ to C **do**
2: Generate momentum variables $p_i' \sim \mathcal{N}(0, M_i)$.
3: Propose moves from (Θ_i', p_i') to $(\Theta_i^\dagger, p_i^\dagger)$ using L leapfrog steps.
4: Calculate $\mathcal{A}_i({\Theta_i}', {\Theta_i}^\dagger) = \min\left\{1, \exp\left(H_i(\Theta_i', p_i') - H_i(\Theta_i^\dagger, p_i^\dagger)\right)\right\}$.
5: Set $\Theta_i^* = \Theta_i^\dagger$ with probability $\mathcal{A}_i({\Theta_i}', {\Theta_i}^\dagger)$, else set $\Theta_i^* = \Theta_i'$.
6: **end for**
7: Simulate $\mathbf{D}^* \sim \mathcal{M}(\mathbf{D} \mid \mathbf{\Theta}^*)$, where $\mathbf{\Theta}^* = \{\Theta_1^*, \Theta_2^*, \cdots, {\Theta_C}^*\}$.
8: Accept $\mathbf{\Theta}^*$ if $\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon_s$, else stay at $\mathbf{\Theta}'$.

Example

This example aims to provide a performance comparison between the two ABC-SS algorithms given a highly diffused prior. Let us assume that the observation $D_{\text{obs}} = 0$ is generated from the Gaussian mixture model $0.5\mathcal{N}\Big(\mu_1(\theta_1, \theta_2), 1\Big) + 0.5\mathcal{N}\Big(\mu_2(\theta_1, \theta_2), 1\Big)$, where

$$\begin{aligned} \mu_1(\theta_1, \theta_2) &= \frac{\theta_1^2}{a^2} + a^2(\theta_2 - b\frac{\theta_1^2}{a^2} - ba^2)^2 - 2\rho\theta_1(\theta_2 - b\frac{\theta_1^2}{a^2} - ba^2) \\ \mu_2(\theta_1, \theta_2) &= (\theta_1 + 1.1)^2 + (\theta_2 - 5)^2 - 3 \end{aligned} \tag{4.27}$$

Equation 4.27 with $a = 1.15$, $b = 0.5$, and $c = 0.9$ is adopted from [143]. Selecting diffuse normal priors $f(\theta_1) = f(\theta_2) = \mathcal{N}(0, 100^2)$ for the parameters, the posterior distribution can be written as

$$f(\theta \mid D_{\text{obs}} = 0) \propto \Big(0.5\mathcal{N}\Big(\mu_1(\theta_1, \theta_2), 1\Big) + 0.5\mathcal{N}\Big(\mu_2(\theta_1, \theta_2), 1\Big)\Big) \times \mathcal{N}(0, 100^2) \times \mathcal{N}(0, 100^2) \tag{4.28}$$

The target posterior distribution is basically a mixture of a banana-shaped distribution and a ring-shaped distribution. The efficacy of each algorithm is determined by how well the samples cover the entire target distribution; this, in fact, will also guarantee a good mixing of the chain samples.

To implement the conventional ABC-SS(MM) algorithm, five simulation levels are

selected with $n_0 = 50000$ and $p_0 = 0.1$. This implies that the accepted samples will have an acceptance probability of 10^{-5}. The proposal standard deviations are selected based on coefficient of variations (COVs) ranging from 0.1 to 0.6. For all six cases with different proposal COVs, the generated samples from the final simulation level are shown on top of the target posterior distribution in Figure 4.8. Similarly, to implement the proposed ABC-SS(MHMC) algorithm, five simulation levels are selected with the same n_0 and p_0 values. A fixed value $L = 10$ for the leapfrog steps is selected along with a varying time step h; where the time step varies between 0.01 and 0.06. The samples obtained from the final simulation level are shown in Figure 4.9.

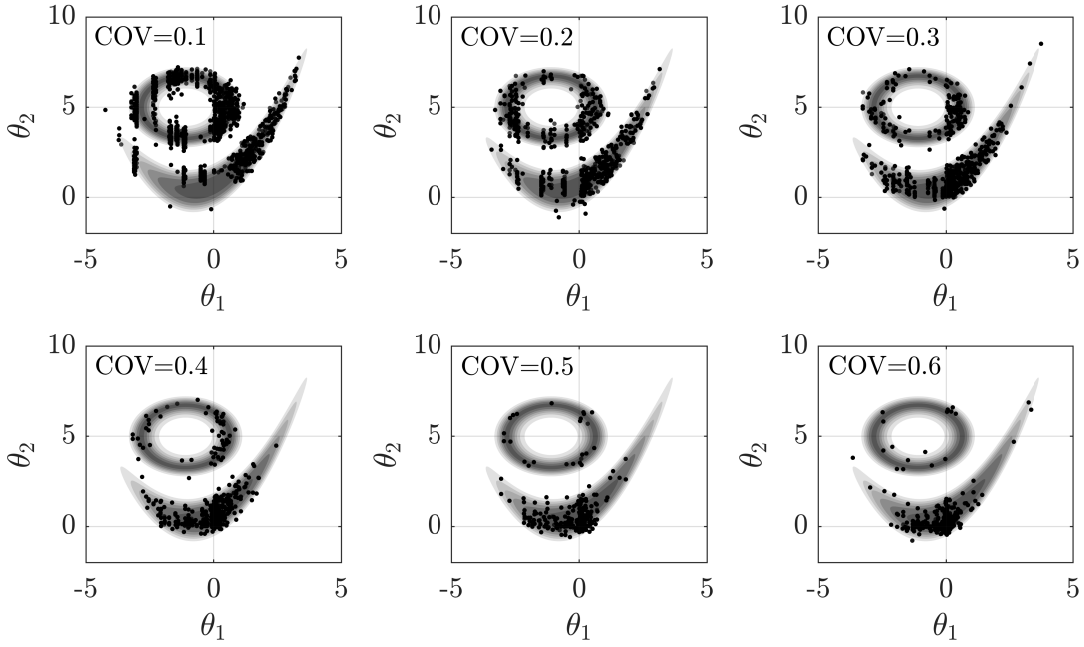


Figure 4.8: The samples (black dots) of the target posterior distribution obtained using the conventional ABC-SS(MM) algorithm with different proposal COVs. The target posterior distribution is shown in gray.

In Figure 4.8, it can be seen that the posterior samples obtained from the ABC-SS(MM) sampler from all six cases do not cover the entire posterior distribution. The samples generated using proposal COVs between 0.1 and 0.3 show high sample repetition and uneven distribution of the samples between the ring-shaped and the banana-

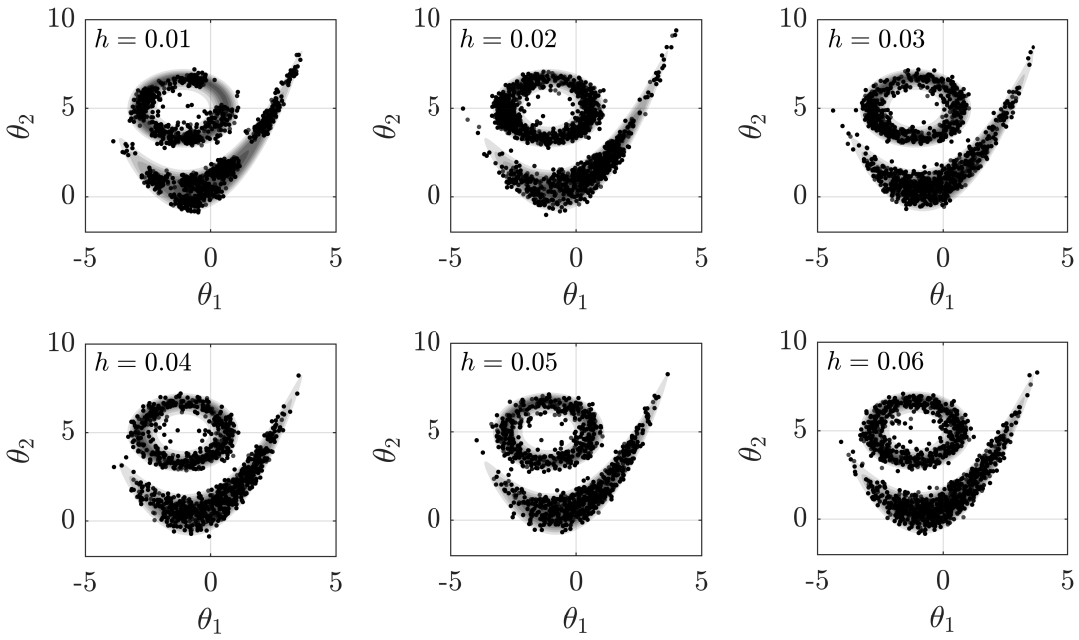


Figure 4.9: The samples (black dots) of the target posterior distribution obtained using the proposed ABC-SS(MHMC) algorithm with a fixed leapfrog step value and different time step values. The target posterior distribution is shown in gray.

shaped distribution. The samples generated using proposal COVs 0.4, 0.5 and 0.6 show less repetition but more concentration on the banana-shaped distribution – indicating a difficulty of the ABC-SS(MM) algorithm in jumping between the two high probability regions. On the other hand, Figure 4.9 shows that the samples obtained using the ABC-SS(MHMC) sampler cover the entire posterior distribution given a higher value for the time step parameter h is chosen. For a small h value such as 0.01, the samples seem to be grouped at different locations of the posterior distribution without covering the entire region. Whereas, for $h \geq 0.03$, the samples show much less repetition and equal distribution between the two high probability regions. This proves that, compared to the conventional ABC-SS(MM) sampler, the proposed ABC-SS(MHMC) sampler has better mixing properties and improved ability to jump between the two high probability regions given a higher time step value is chosen.

4.10 Model Selection using ABC

Bayesian model selection is a model selection tool that uses the Bayes factor to compare between any two given models [72]. To understand how it can be implemented in practice, suppose $\mathcal{M}_1$ and $\mathcal{M}_2$ are two models with prior distributions (or probabilities) $f(\mathcal{M}_1)$ and $f(\mathcal{M}_2)$, respectively. Once the observed data $\mathbf{D}_{\text{obs}}$ are available, the prior distributions can be updated to the corresponding posterior distributions as

$$f(\mathcal{M}_i \mid \mathbf{D}_{\text{obs}}) \propto f(\mathbf{D}_{\text{obs}} \mid \mathcal{M}_i) f(\mathcal{M}_i), \quad i = 1, 2 \tag{4.29}$$

Using Equation 4.29, one can find the ratio between the two posteriors as

$$\frac{f(\mathcal{M}_1 \mid \mathbf{D}_{\text{obs}})}{f(\mathcal{M}_2 \mid \mathbf{D}_{\text{obs}})} = \frac{f(\mathbf{D}_{\text{obs}} \mid \mathcal{M}_1) f(\mathcal{M}_1)}{f(\mathbf{D}_{\text{obs}} \mid \mathcal{M}_2) f(\mathcal{M}_2)} \tag{4.30}$$

The Bayes factor $\mathcal{B}_{12}$ is given by the ratio between the two likelihoods, i.e.,

$$\mathcal{B}_{12} = \frac{f(\mathbf{D}_{\text{obs}} \mid \mathcal{M}_1)}{f(\mathbf{D}_{\text{obs}} \mid \mathcal{M}_2)} = \frac{f(\mathcal{M}_1 \mid \mathbf{D}_{\text{obs}}) / f(\mathcal{M}_2 \mid \mathbf{D}_{\text{obs}})}{f(\mathcal{M}_1) / f(\mathcal{M}_2)} \tag{4.31}$$

Thus, the Bayes factor is essentially a ratio of the posterior odds of a particular model to its prior odds [72], where odds can be defined as (probability/1 − probability). In other words, it is a quantitative measure that provides the goodness-of-fit or the evidence in support of a particular model over another. Kass and Raftery [72] provided an interpretation of the Bayes factor, which can be described as follows. In support of model $\mathcal{M}_1$ and against model $\mathcal{M}_2$, $3 \geq \mathcal{B}_{12} > 1$ shows very weak evidence, $20 \geq \mathcal{B}_{12} > 3$ shows positive evidence, $150 \geq \mathcal{B}_{12} > 20$ shows strong evidence, and $\mathcal{B}_{12} > 150$ shows very strong evidence. This interpretation helps ABC methods to choose the model that best represents the data from a collection of available models.

The model selection algorithm using ABC-SS(MHMC) is a natural extension to its parameter estimation version. Here, we present a Bayesian model selection framework by adding an extra parameter, the model index $m = 1, 2, \cdots, n_m$, to the proposed algorithm. Parameters of model $\mathcal{M}_m$ are denoted as $\mathbf{\Theta}_m = \{\theta_{m,1}, \theta_{m,2}, \cdots, \theta_{m,d_m}\}$, where d_m is the dimension of the parameter space of the respective model. The ABC-SS(MHMC) algorithm for model selection proceeds as follows:

Algorithm 12 ABC-SS(MHMC) sampler for model selection

1: At simulation level $s = 1$, generate $\{\mathcal{M}_m^{(k)}, \mathbf{\Theta}_m^{(k)}, \mathbf{D}^{(k)}\}_{k=1}^{n_0} \overset{\text{iid}}{\sim} \mathcal{M}(\mathbf{D} \mid \mathbf{\Theta}) f(\mathbf{\Theta}) f(\mathcal{M})$.
2: Calculate $\rho^{(k)} = \rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}})$, and renumber as $\rho^{(1)} < \rho^{(2)} < \cdots < \rho^{(n_0)}$.
3: Renumber $\{\mathbf{\Theta}_m^{(k)}\}_{k=1}^{n_0}$ accordingly and set $\epsilon_s = \rho^{(n_0 p_0)}$.
4: Use $\{\mathbf{\Theta}_m^{(k)}\}_{k=1}^{n_0 p_0}$ as seed samples to run Algorithm 11 with tolerance ϵ_s.
5: Repeat steps 2-4 until simulation level s reaches S.

At the first simulation level, models are first sampled from the prior $f(\mathcal{M})$, then the model-specific parameters are sampled from their respective prior distributions $f(\mathbf{\Theta}_m \mid \mathcal{M})$. The next step is to simulate the corresponding data sets as $\mathbf{D} \sim \mathcal{M}(\mathbf{D} \mid \mathbf{\Theta}_m)$ and calculate the corresponding distances. Finally, the $p_0 n_0$ smallest distance values and their corresponding samples are identified. These samples are then kept for the next level, and the rest of the samples are discarded. At other simulation levels, seed samples from specific models are used in MHMC to generate new parameters independently. Thus, multiple MHMC chains are created but different chains correspond to different models depending on the seed sample. Similar to the parameter estimation process, the distances $\rho(\mathbf{D}, \mathbf{D}_{\text{obs}})$ are calculated, and the $n_0 p_0$ smallest distances are identified. Finally, the parameter samples corresponding to the $n_0 p_0$ identified distances are kept as seed samples for the next simulation level, and the rest are discarded. During this process, samples from specific models may completely vanish if the underlying model provides poor fit to the data.

The output samples of the models belong to the marginal posterior distribution

$f(\mathcal{M} \mid \mathbf{D}_{\text{obs}})$, whereas the output model-specific parameter samples belong to the marginal posterior distribution $f(\boldsymbol{\Theta} \mid \mathcal{M}, \mathbf{D}_{\text{obs}})$. The model that has the largest number of parameter samples is considered the best fit for the given data. The posterior probabilities of the models can be obtained using the number of parameter samples of individual models divided by the total number of parameter samples available. The posterior probabilities, then, can be used to calculate the Bayes factors using Equation 4.31, so that the models under consideration can be compared on how well they fit with the data at hand.

Algorithm 12 proceeds simultaneously with the model selection and parameter estimation in an intertwined fashion. The advantage of this is that the output will contain samples from several models, and these samples can be used in Bayesian model averaging which is build upon the idea that the average of the underlying models may sometimes fit the data better than a single model [62]. A real-life example demonstrating the application of the proposed model selection algorithm is presented in Chapter 6.

4.11 Concluding Remarks

This chapter presented the basic idea behind the likelihood-free inference approach and derivations of a variety of ABC algorithms with varying efficiency and complexity. Two new algorithms are proposed – one is based on the HMC sampling scheme and the other is based on the subset simulation scheme.

The HMC sampling scheme follows the Hamiltonian dynamics to propose new samples from seed samples. Its non-random walk behavior helps to explore the target probability space more effectively and efficiently than the standard random-walk MCMC method. The standard ABC-MCMC scheme suffers from poor mixing of samples which leads to a very high number of sample repetitions in a Monte Carlo Markov chain (for

example, see [61]). The HMC sampling scheme, on the contrary, provides better mixing properties when integrated with ABC. The example provided in Section 4.9.2 proved this fact by showing that the new ABC-HMC scheme captures the small mode of a target posterior distribution more effectively than the standard ABC-MCMC sampling scheme.

The proposed ABC-HMC scheme is further integrated with a rare event sampler, subset simulation, which significantly increases its efficiency. Moreover, a modified HMC sampler is introduced, which is essentially the component-wise version of the likelihood-free HMC sampler. With faster convergence, the new ABC-SS(MHMC) sampler turns out to be a powerful method to sample from a complex multi-modal target density. The proposed algorithm showed promising performance in the example in Section 4.9.3, where the algorithm successfully explored the target probability space using a highly diffused prior. Whereas, the standard ABC-SS(MM) scheme struggled to capture the entire high probability region under a similar setting.

The applicability of the proposed ABC-SS(MHMC) sampling scheme is further extended by transforming it into a likelihood-free Bayesian model selection tool. The proposed model selection algorithm selects the best model from a set of available models using the Bayes factors. To demonstrate the applicability of the likelihood-free ABC methods in degradation modeling, practical real-life examples are presented in the next two chapters.

Chapter 5

Parameter Estimation of Corrosion Growth Models

5.1 Introduction

The application of the proposed likelihood-free methods is demonstrated in this chapter by using FAC growth models. FAC is a life-limiting factor for the feeder piping network of the primary heat transport system of CANDU® reactors [33, 121]. The feeder pipes contain pressurized heavy-water coolant which carries the heat generated in the reactor core to the steam generators. Figure 5.1 shows the typical layout of the feeder pipe assembly on the reactor face. Each fuel channel in the reactor is connected to an inlet and an outlet feeder pipe. The number of feeder pipes may vary between 380 and 480 depending on the reactor type. The diameters of outlet feeder pipes vary between 2 inches (51 mm) and 2.5 inches (64 mm) with nominal wall thicknesses varying between 5.5 mm and 7.0 mm, respectively [94].

FAC can be described as a process where the protective coating of oxide layer on carbon steel gets dissolved in a stream of flowing water [42]. FAC is essentially an

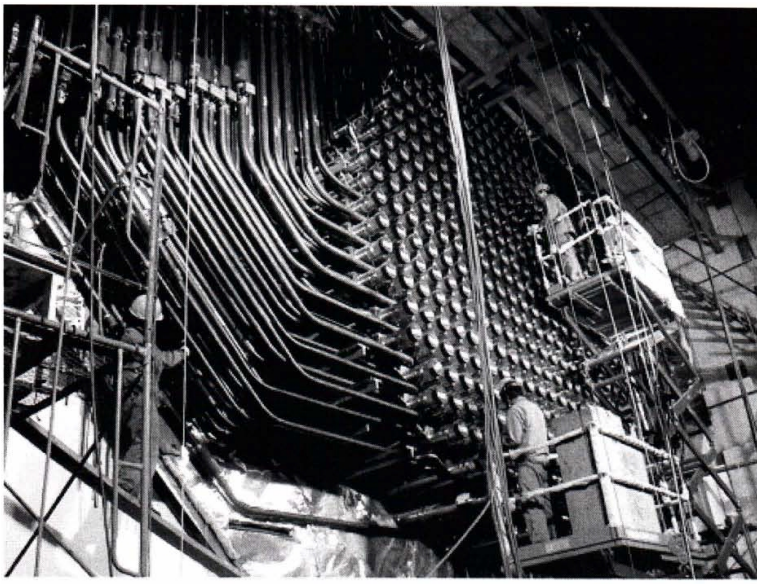


Figure 5.1: Feeder pipe assembly on the CANDU® reactor face [47]. (Image reproduced with permission.)

electrochemical corrosion, which is governed by the process of mass transfer in flowing water [147]. More discussions on the physical-chemical mechanisms of the FAC process can be found in several references [14, 15, 24, 25, 44]. The FAC process is mainly governed by high flow velocities (>15 m/s), high pH (>10), and high temperatures (>300°C) during plant operations [60]. The wall thickness losses are generally higher at the tight radius bends and the welded joints compared to other sections of the pipes. Since the FAC process is dependent on several factors such as the chemical environment, temperature, flow turbulence, and the geometry of the pipes, different feeder pipes of the primary heat transport system experience different rates of wall thinning [100].

The wall thickness of a specific feeder pipe experiencing FAC at time t can be represented as

$$W(t) = w_0 - X(t) \tag{5.1}$$

where w_0 is the nominal pipe wall thickness and $X(t)$ is the wall thickness loss due

to FAC in the time interval 0 to t. To ensure the fitness-for-service of a pipe, a minimum wall thickness limit $W_{\min}$ is specified. A feeder pipe section is considered substandard when $W(t) \leq W_{\min}$. The time at which $W(t)$ reaches $W_{\min}$ indicates the end of life of the pipe section. The wall thickness loss $X(t)$ can be characterized using a probabilistic/stochastic degradation model.

The next sections present three examples based on the FAC data collected from the feeder pipes. Various advanced likelihood-free schemes are implemented for the parameter inference task.

5.2 Example I: Estimation of the Distribution of FAC Rate

For monitoring the extent of degradation and predicting the lifetime of the feeder pipes, it is often needed to accurately estimate the wall thinning rate due to FAC in the piping components. The simplest way to estimate the FAC rate is to model the degradation data using the linear random rate model (for details, see Subsection 2.2.1). In this model, the FAC rate is assumed to be a random variable that follows a specified probability distribution with unknown parameters. This assumption is justified by the fact that different pipes experience different rates of wall thinning due to variable operating conditions. To estimate the parameters of the distribution of FAC rate, the ABC-MCMC method (Algorithm 7) has been implemented. ABC-MCMC is compared with the traditional MH algorithm (Algorithm 2), and the results from both MCMC methods are further compared with the results from the linear regression analysis – a common industrial approach for the FAC rate estimation. In this study, the term likelihood-based MCMC (L-MCMC) is equivalently used to indicate the MH algorithm unless otherwise stated.

5.2.1 Degradation Data and Model

Data Set

The inspection data related to the minimum wall thicknesses of 37 feeder pipes are considered in this study. These pipes have a common diameter of 2" and nominal thickness of 5.5 mm. Figure 5.2 shows the degradation paths of the feeder pipes, i.e., each gray line represents the wall thickness losses over time of a different pipe derived from the minimum wall thickness measurements. These feeder pipes were inspected two to five times using an ultrasonic probe between 8.55 and 22.25 effective full power years (EFPY).

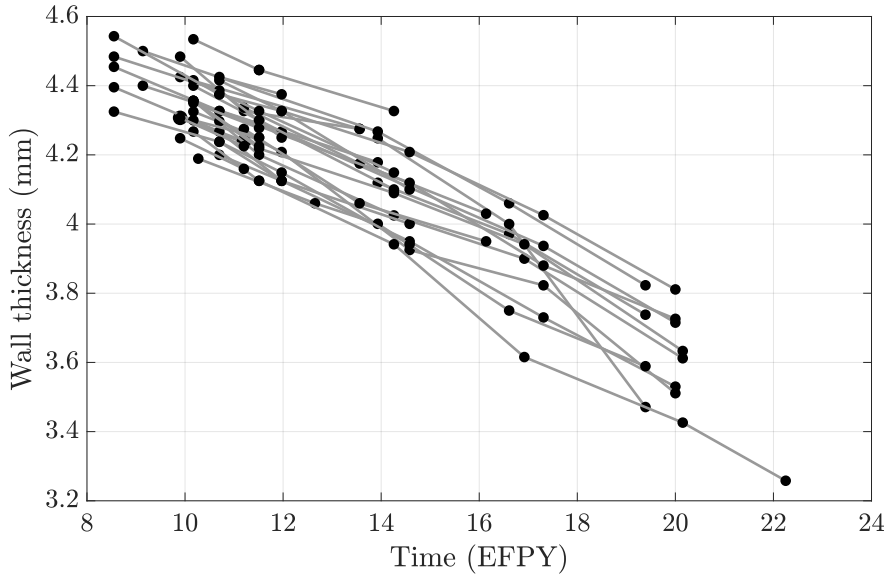


Figure 5.2: Wall thickness measurements of the feeder pipes.

Model

The random rate model (Subsection 2.2.1) is used to characterize the wall thinning data. The wall thinning rate R is assumed to be a gamma distributed random variable with PDF

$$f_R(r) = \frac{r^{\mu_R^2/\sigma_R^2} - 1}{(\sigma_R^2/\mu_R)^{\mu_R^2/\sigma_R^2}\Gamma(\mu_R^2/\sigma_R^2)} \exp(-r\mu_R/\sigma_R^2) \tag{5.2}$$

where $\mu_R > 0$ and $\sigma_R > 0$ are the mean and standard deviation of the FAC rate R, respectively. Without the loss of generality, Equation 5.2 represents the gamma PDF with alternative parameters, mean μ_R, and standard deviation σ_R, instead of the shape and scale parameters for ease of interpretation. It is assumed that μ_R is unknown, whereas σ_R has a fixed value of 0.01 mm/EFPY. The initial wall thickness A is modeled as a normally distributed random variable with the PDF $f_A(a) = \mathcal{N}(\mu_A, \sigma_A^2)$. Again, mean μ_A is assumed to be unknown and σ_A is assumed to have a fixed value of 0.1 mm. The inspection data, collected using an ultrasonic probe, are assumed to be contaminated by a normally distributed sizing error Z with zero mean and 0.1 mm standard deviation.

5.2.2 Implementation Details

The Bayesian inference schemes, ABC-MCMC and L-MCMC, and the linear regression method are implemented in the MATLAB environment (version 9.3, 64-bit) with Intel® Core™ i5-6500 CPU @3.20 GHz processor and 8.00 GB RAM memory.

Prior and Proposal Distributions

For implementing the ABC-MCMC scheme, the first step is to select a prior distribution. Assuming independence, one can conveniently select $f(\boldsymbol{\Theta}) = f_1(\mu_A)f_2(\mu_R)$, where $\boldsymbol{\Theta} = \{\mu_A, \mu_R\}^\top$. The chosen prior parameter distributions are assumed to be uniform as $f_1(\mu_A) = \mathcal{U}[3.5, 7.5]$ and $f_2(\mu_R) = \mathcal{U}[0, 2]$. Note that an uninformative uniform prior assigns equal probabilities to all the possible values in a specified range. Next, proposal distributions are chosen to be normal and log-normal distributions with means equal to

the current samples and standard deviations equal to 0.02 mm and 0.002 mm/EFPY for μ_A and μ_R, respectively. The main reason to select a log-normal proposal density for μ_R is that μ_R represents the mean of the gamma distribution, which is, by definition, a strictly positive quantity. Thus, it is required that the chosen proposal density does not generate negative μ_R values. To implement the L-MCMC or MH algorithm, the same prior distributions are chosen as used in ABC-MCMC. However, the proposal distributions are chosen to be normal and log-normal distributions with means equal to the current samples and standard deviations equal to 0.05mm and 0.005 mm/EFPY for the parameters μ_A and μ_R, respectively. The likelihood function is numerically integrated using the Monte Carlo simulation method with 1000 samples.

Distance Function and Tolerance Threshold

The Euclidean distance is generally considered to be a suitable distance measure for data-vectors. Therefore, it is a good choice for a distance function for the ABC-MCMC scheme (see Section 4.5). Accordingly, the selected distance function can be written as

$$\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{y}_i^* - \mathbf{y}_i\| \tag{5.3}$$

where $\| \bullet \|$ is the Euclidean norm (or ℓ_2 norm) operator and $\mathbf{y}_i^*$ represents simulated data. The distance function is an average of the Euclidean norm of the difference between the observed data and the simulated data from each pipe component. On the other hand, instead of choosing a single threshold level, a series of threshold values in a descending order, i.e., $\epsilon = \{0.5, 0.4, 0.3, 0.25\}$, are chosen to observe the convergence of the approximate posteriors. The threshold value $\epsilon = 0.25$ can be considered to be sufficiently small given that further reduction of this value may cause a very high rejection rate, making the algorithm computationally prohibitive.

A New Initialization Scheme for ABC-MCMC

To reduce the initialization time of the MCMC chain, we bring an extra step to the ABC-MCMC algorithm. The MCMC chain generally starts with a random value generated from the prior distribution and allowed to burn-in until it converges to its stationary distribution. But, if ϵ is very small, then the burn-in period becomes very long, making the algorithm computationally expensive. To avoid burn-in, the basic ABC-RS can be used to initialize the MCMC chain. However, the same low acceptance problem due to small ϵ prevails in this case as well. But, the advantage of using the ABC-RS to initialize the MCMC chain is very clear, because, as soon as the first sample directly comes from the target distribution, the burn-in stage is not required anymore. Thus, the proposed algorithm starts with the selection of a series of tolerance thresholds $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_T$ (ϵ_T is the target tolerance threshold), and sequentially generates the corresponding set of samples $\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Theta}^{(2)}, \cdots, \boldsymbol{\Theta}^{(T)}$. The initial samples $\boldsymbol{\Theta}^{(1)}, \cdots, \boldsymbol{\Theta}^{(T-1)}$ are discarded, and the ABC-MCMC algorithm starts with $\boldsymbol{\Theta}^{(T)}$ as its initial sample. Algorithm 13 shows the steps to implement the proposed algorithm. A

Algorithm 13 Initialization scheme

1: Select a series of tolerance thresholds $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_T$
2: Generate $\boldsymbol{\Theta}^{(1)} \sim f_{\epsilon_1}(\boldsymbol{\Theta} \mid \mathbf{D}_{\text{obs}})$ using ABC-RS
3: **for** $k = 1$ to $T - 1$ **do**
4: **repeat**
5: Generate $\boldsymbol{\Theta}^* \sim q'(\boldsymbol{\Theta}^* \mid \boldsymbol{\Theta}^{(k)})$, such that $f(\boldsymbol{\Theta}^*) \neq 0$
6: Simulate $\mathbf{D}^* \sim \mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}^*)$
7: Accept $\boldsymbol{\Theta}^*$, if $\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) \leq \epsilon_k$
8: **until** Acceptance
9: Set $\boldsymbol{\Theta}^{(k+1)} = \boldsymbol{\Theta}^*$
10: **end for**
11: Discard $\boldsymbol{\Theta}^{(1)}, \cdots, \boldsymbol{\Theta}^{(T-1)}$ and start ABC-MCMC using $\boldsymbol{\Theta}^{(T)}$.

suitable COV for the initialization proposal distribution $q'(\bullet \mid \bullet)$ should be chosen so that the algorithm does not get stuck, i.e., too narrow or too wide support for the pro-

posal distribution should be avoided. To initialize ABC-MCMC, the chosen threshold values are, $\epsilon_i = n^* \times \epsilon_T$, $i = 1, 2, \cdots, 6$, where $n^* = \{1.5, 1.4, 1.3, 1.2, 1.1, 1\}$. Similarly, the proposal density is considered to be the same normal and log-normal product densities with means equal to the current samples and standard deviations equal to 0.04 mm and 0.004 mm/EFPY for μ_A and μ_R, respectively.

Linear Regression

The steps for linear regression analysis of the wall thickness measurement data are described as follows. The wall thickness measurements can be modeled using a linear regression model, $y_{ij} = \beta_{i0} + \beta_{i1} t_{ij} + z_i$, where β_{i0}, β_{i1}, and z_i are the intercept, slope, and random error of regression, respectively. The intercept β_{i0} denotes initial wall thickness a_i of the ith feeder pipe, whereas the absolute value of the slope or the regression coefficient, i.e., $|\beta_{i1}|$, represents the FAC rate or the wall thinning rate r_i of the same pipe. To obtain the parametric forms of the distributions of FAC rates and initial wall thicknesses, the regression coefficients $|\beta_{11}|, |\beta_{21}|, \cdots, |\beta_{N1}|$ and the intercepts $|\beta_{10}|, |\beta_{20}|, \cdots, |\beta_{N0}|$ are fitted to a gamma and a normal distribution, respectively, using the MLE method.

5.2.3 Results and Discussion

Parameter Estimates

The ABC-MCMC method, for all four cases, i.e., $\epsilon = \{0.5, 0.4, 0.3, 0.25\}$, took around 132 s, 124 s, 125 s, and 125 s, respectively, to generate Markov chains of lengths equal to 2.5×10^5 samples. The first five samples are discarded for the initialization process. Figure 5.3 shows the probability density estimates of the distance function generated from the ABC-MCMC algorithm for all four cases. It can be noticed that the value

$\epsilon = 0.25$ belongs to the left tail portion of its corresponding density estimate, implying that $\rho < \epsilon = 0.25$ is a rare event compared to the other three cases of ABC-MCMC. By

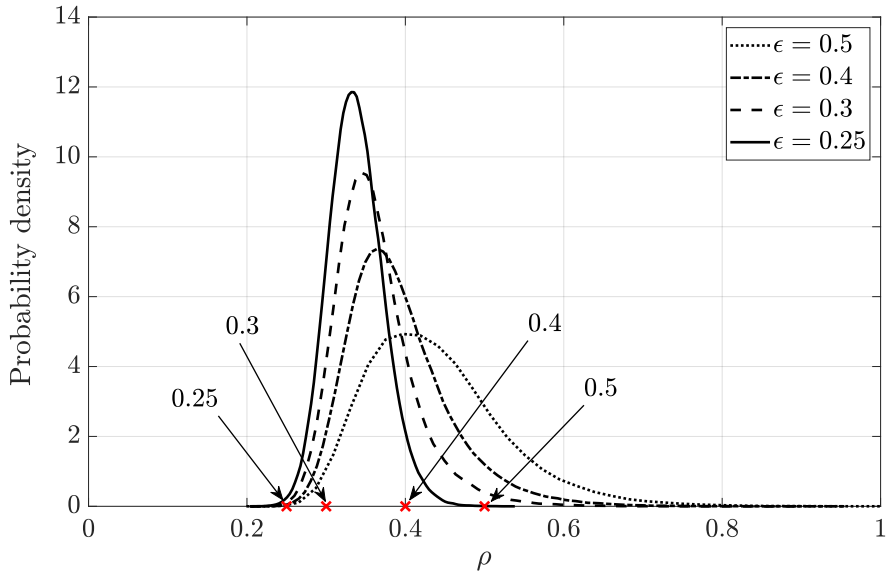


Figure 5.3: Probability density estimates of the distance function values for all four cases of the ABC-MCMC algorithm.

comparison, the L-MCMC method took around 494 s on the same computer to generate a Markov chain of length equal to 5000 samples; the initial 500 samples are rejected to allow for burn-in. The Markov chains generated from both methods are plotted in Figure 5.4. It is evident from the figure that the Markov chains of ABC- MCMC and L-MCMC schemes quickly converged to their stationary distributions. The Markov chain generated from the ABC-MCMC scheme with $\epsilon = 0.25$ shows lowest variance with respect to the other Markov chains generated from the same scheme.

Figure 5.5 presents the two-dimensional scatter plots of the accepted samples from ABC-MCMC and L-MCMC algorithms. The figure illustrates clearly the convergence of the joint posteriors of μ_A and μ_R with respect to the tolerance thresholds computed using ABC-MCMC. It is clear that the joint posterior of $\{\mu_A, \mu_R\}^\top$, obtained from

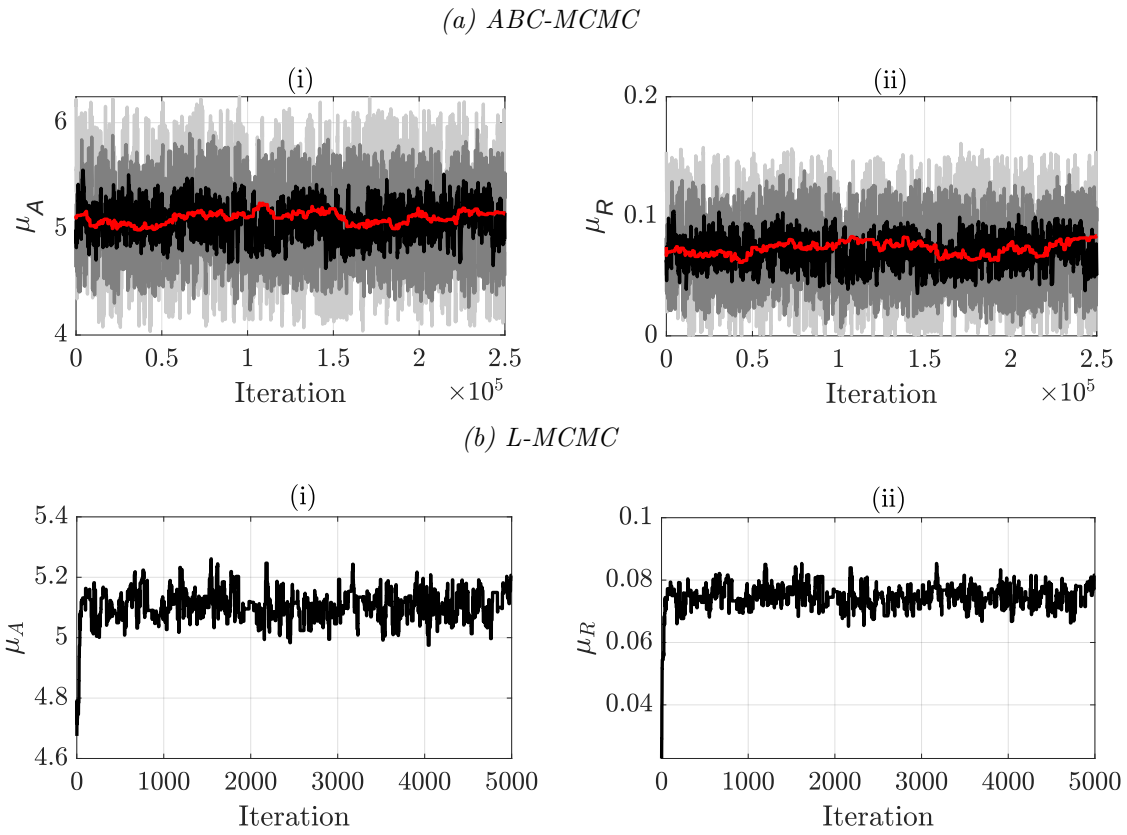


Figure 5.4: Markov chains of parameters (i) μ_R and (ii) μ_A generated from (a) ABC-MCMC and (b) L-MCMC samplers. In subfigure (a), the light gray line corresponds to the Markov chain produced by ABC-MCMC with $\epsilon = 0.5$, the deep gray line corresponds to $\epsilon = 0.4$, the black line corresponds to $\epsilon = 0.3$, and the red line corresponds to $\epsilon = 0.25$.

ABC-MCMC with $\epsilon = 0.25$, lies arbitrarily close to the joint posterior obtained from the L-MCMC algorithm.

Figure 5.6 and Figure 5.7 show the marginal posteriors of the FAC model parameters μ_A and μ_R, respectively. Again, the convergence of the ABC posteriors with respect to the threshold values are clearly visible. The marginal posteriors of μ_A and μ_R generated from ABC-MCMC algorithm with $\epsilon = 0.25$ and L-MCMC show great similarity. The statistical properties, i.e., mean, COV, and 90% CI, of the marginal posteriors are presented in Table 5.1.

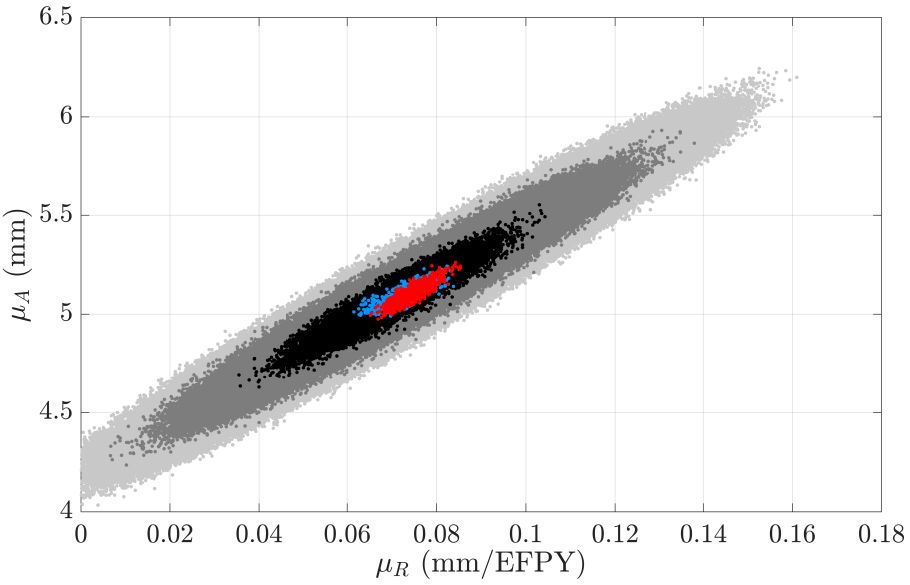


Figure 5.5: The accepted posterior samples of the ABC-MCMC and L-MCMC algorithms as two-dimensional scatterplots. The light gray dots correspond to the posterior samples obtained from ABC-MCMC with $\epsilon = 0.5$, the deep gray dots correspond to $\epsilon = 0.4$, the black dots correspond to $\epsilon = 0.3$, and the blue dots correspond to $\epsilon = 0.25$. The red dots correspond to the posterior samples obtained from L-MCMC.

Table 5.1: Statistical properties of the marginal posteriors.

Parameter	ABC-MCMC ($\epsilon = 0.25$)			L-MCMC		
	Mean	COV	[5th, 95th] percentiles	Mean	COV	[5th, 95th] percentiles
μ_A (mm)	5.10	0.011	[5.01, 5.19]	5.11	0.010	[5.02, 5.19]
μ_R (mm/EFPY)	0.073	0.074	[0.064, 0.082]	0.075	0.045	[0.069, 0.081]

Results from Linear Regression Analysis

Figure 5.8 and Figure 5.9 present the histogram plots of the initial wall thickness A and the FAC rate R, respectively, obtained using the linear regression analysis. The corresponding normal and gamma distribution fits are also plotted on top of the histograms. The statistical properties of A and R are shown in Table 5.2. The 95th percentile of R, i.e., R_{95}, is calculated from the fitted gamma distribution and found to be around 0.099 mm/EFPY. It is to be remarked that R_{95} represents the 95%

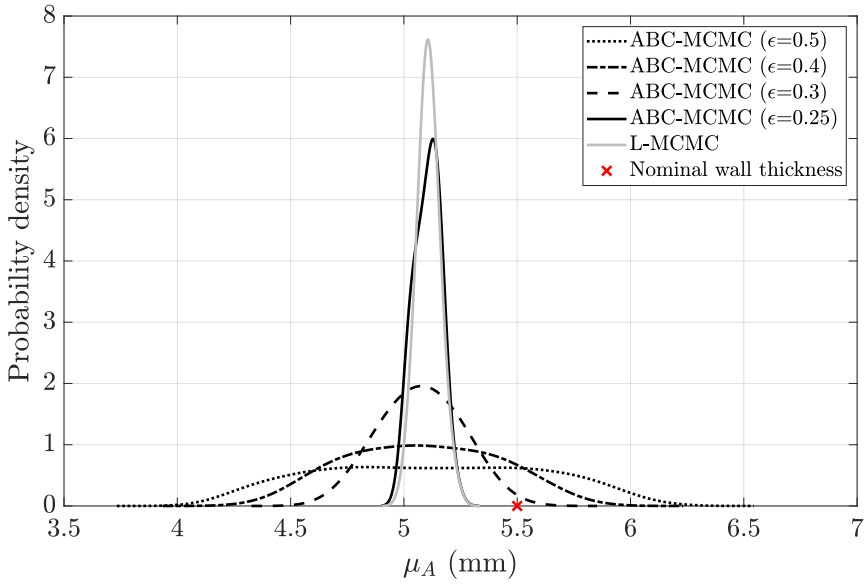


Figure 5.6: Marginal posterior distribution of mean μ_A of the initial pipe wall thickness.

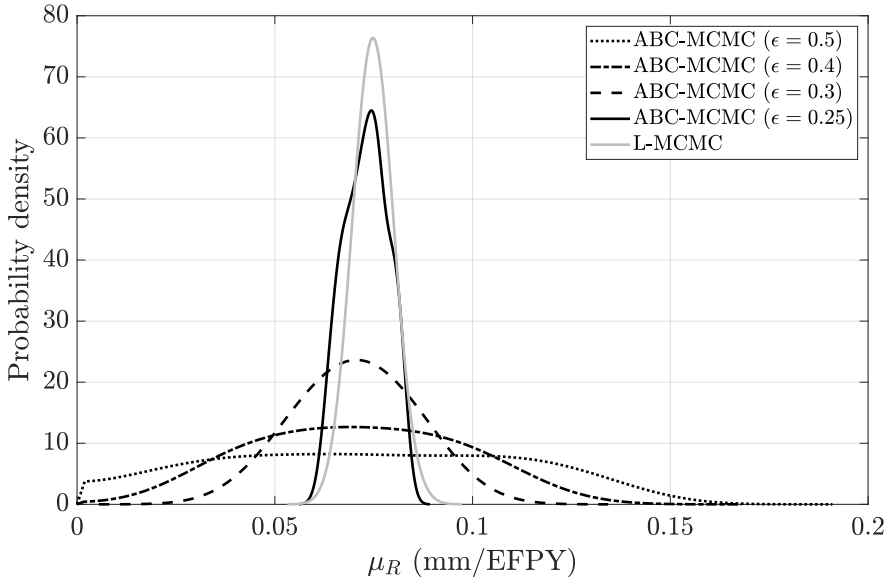


Figure 5.7: Marginal posterior distribution of mean μ_R of the FAC rate.

probability of non-exceedance of the FAC rate, and therefore its estimation plays a decisive role in maintenance planning.

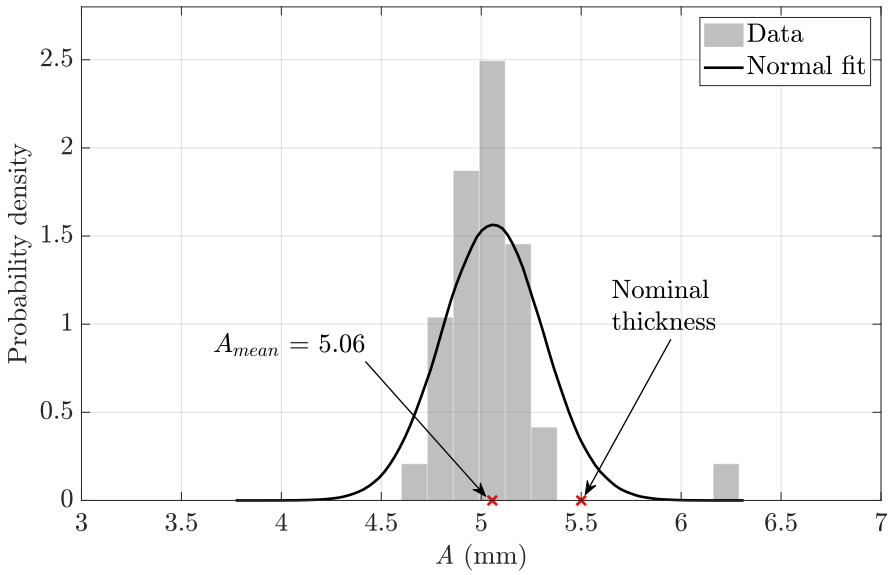


Figure 5.8: Histogram of the initial wall thickness A along with a normal fit obtained from the linear regression analysis.

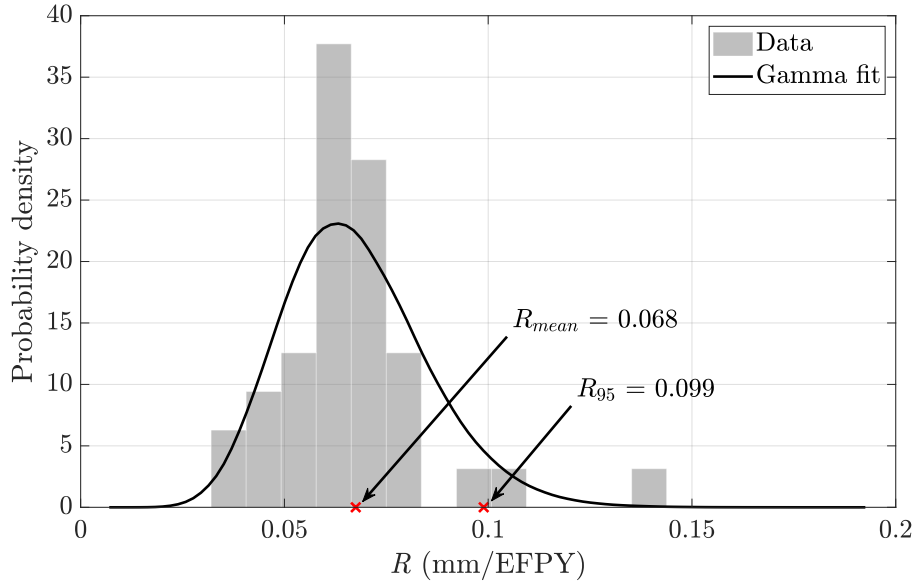


Figure 5.9: Histogram of the FAC rate R along with a gamma fit obtained from the linear regression analysis.

Table 5.2: Results of linear regression analysis.

Parameter	Mean	COV	[5th, 95th] percentiles
A (mm)	5.06	0.050	[4.64, 5.47]
R (mm/EFPY)	0.068	0.261	[0.041, 0.099]
T (EFPY)	45.41	0.296	[27.99, 70.32]

95th Percentile FAC rate and Lifetime Distribution

In the Bayesian framework, the R_{95} is treated as a random quantity that has a distribution associated with it, and the ensuing quantity of interest is the 95th percentile of R_{95}, also known as $R_{95/95}$, i.e., the 95/95 value of the FAC rate. The probability density estimates of R_{95} (see Table 5.3 for statistical properties) obtained from ABC-MCMC and L-MCMC are plotted in Figure 5.10. Both the algorithms produce similar density estimates for the R_{95} value. The ABC-MCMC and L-MCMC algorithms give very close estimates of the 95/95 value of the FAC rate, which are around 0.099 mm/EFPY and 0.098 mm/EFPY, respectively. As shown in the figure, the R_{95} estimate obtained from the linear regression analysis found to be very close to the $R_{95/95}$ estimates of the ABC-MCMC and L-MCMC algorithms.

Table 5.3: Statistical properties of the distributions of R_{95} and T_{05}.

Quantity	ABC-MCMC ($\epsilon = 0.25$)			L-MCMC		
	Mean	COV	[5th, 95th] percentiles	Mean	COV	[5th, 95th] percentiles
R_{95} (mm/EFPY)	0.090	0.059	[0.081, 0.099]	0.092	0.036	[0.086, 0.098]
T_{05} (EFPY)	32.01	0.046	[29.55, 34.68]	31.38	0.023	[30.25, 32.60]

To determine the lifetime distribution of the feeder pipes, the minimum wall thickness requirement $W_{\min}$ is considered to be 40% of the nominal thickness, i.e., $W_{\min} = 0.4 \times w_0$. In Figure 5.11, the lifetime distribution of the feeder pipes obtained from the linear regression analysis is plotted, and Table 5.2 lists its statistical properties. Another pertinent quantity of interest in the context of feeder lifetime analysis is the fifth percentile of lifetime T_{05}, and in this analysis, it is estimated to be around 27.99 EFPY.

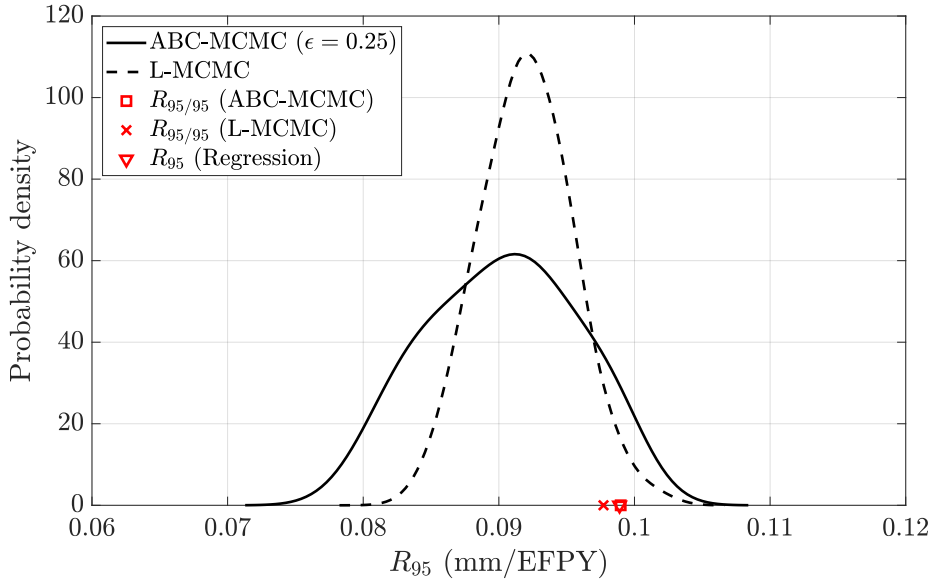


Figure 5.10: Probability density estimate of the 95th percentile values of the FAC rate obtained using ABC-MCMC and L-MCMC algorithms.

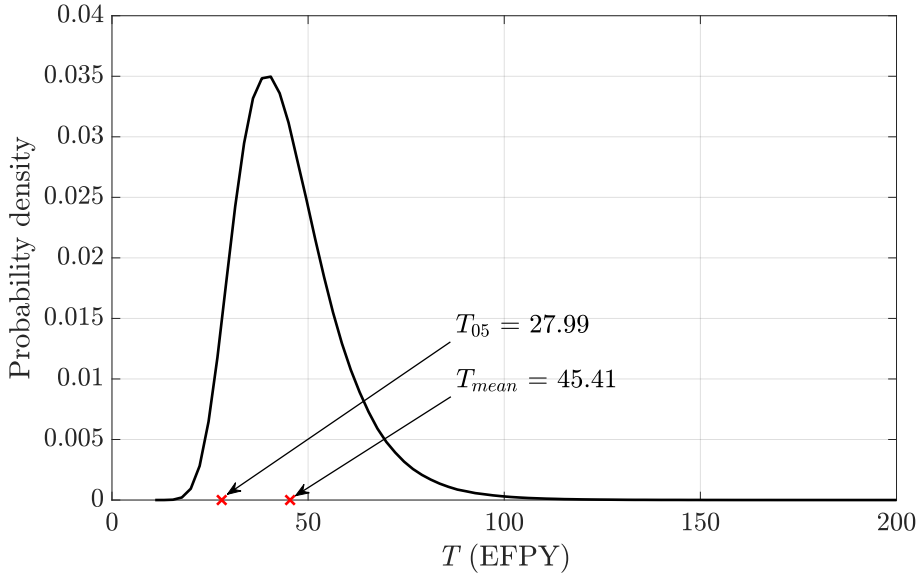


Figure 5.11: Probability density estimate of the lifetime of feeder population obtained from linear regression analysis.

Figure 5.12 presents the probability density estimates of T_{05} derived from ABC-MCMC and L-MCMC. Again, it can be observed that both Bayesian computation algorithms produce similar density estimates for T_{05}. The distribution characteristics of T_{05} can be found in Table 5.3. Similar to $R_{95/95}$, the 05/05 value of the population lifetime $T_{05/05}$ (fifth percentile of T_{05}), indicate the 95% probability of exceedance in 95% of the feeder pipes. The estimated $T_{05/05}$ values using ABC-MCMC and L-MCMC are found to be around 29.55 EFPY and 30.25 EFPY, respectively. The $T_{05/05}$ estimates are very close, showing strong competitiveness between both algorithms. The T_{05} estimate of the linear regression analysis can be found to be lower than the $T_{05/05}$ estimates of the ABC-MCMC and L-MCMC algorithms, displaying an underestimation of the feeder lifetime by the regression approach.

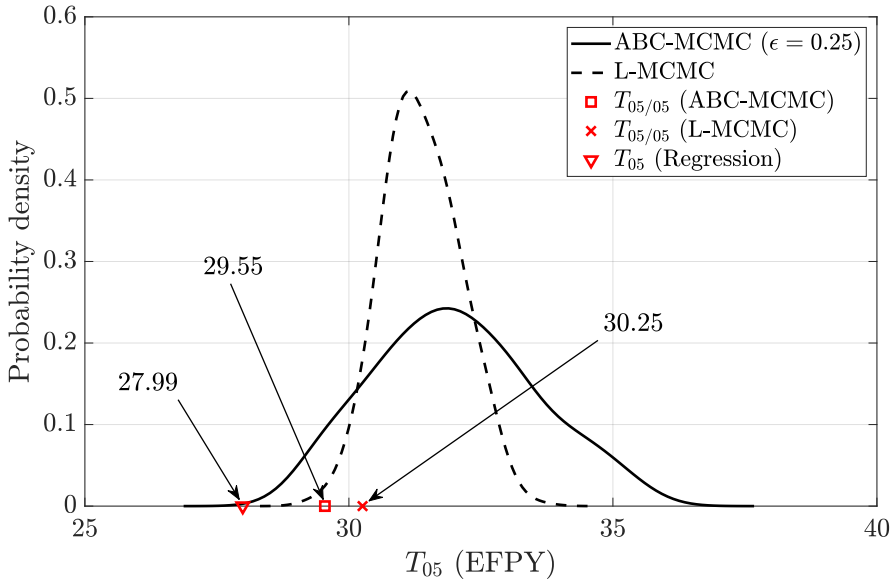


Figure 5.12: Probability density estimate of the fifth percentile of the lifetime of feeder population obtained using ABC-MCMC and L-MCMC algorithms.

However, it may be noticed in Figure 5.10 and Figure 5.12 that the density estimates of R_{95} and T_{05} from ABC-MCMC show slightly higher variances when compared to

the L-MCMC estimates (Table 5.3). This occurs due to the approximation in the ABC posterior introduced by the selection of the tolerance threshold. As mentioned before, the tolerance threshold ϵ acts as a trade-off between the computational efficiency and accuracy, and therefore selecting a smaller value of ϵ can produce more accurate posterior variances (closer to the posterior variances from L-MCMC) albeit this comes at a higher computational cost.

5.3 Example II: Estimation of Gamma Process Parameters from Noisy Data

The gamma process is a popular stochastic process model for characterizing a wide variety of degradation processes affecting engineering structures and components. Although the conceptual approach to the parameter estimation of the gamma process is straightforward, its practical implementation is quite challenging since degradation data are often contaminated by measurement noise. As a result, the likelihood function for sample data turns into a high-dimensional multivariate integral (refer to Subsection 2.2.2 to see the form of the noisy likelihood). This example presents the application of the likelihood-free ABC-MCMC method (Algorithm 7) for the parameter estimation task. To investigate and compare the efficacy of the method with the standard likelihood-based approach, various simulation examples on the estimation of the gamma process parameters are presented in Appendix D. The simulation examples show that the results of ABC-MCMC are comparable with the results of L-MCMC, while offering significant computational savings.

5.3.1 Degradation Data and Model

Data Set

This example considers the same data set used in the first example in Section 5.2, which is related to the minimum wall thicknesses of 37 feeder pipes.

Model

The wall thickness data are first converted to the wall thickness losses over time intervals, and then the gamma process model is used to characterize the data. The basic properties of the gamma process and details of the derivation of the likelihood function under noisy data can be found in Subsection 2.2.2.

5.3.2 Implementation Details

The ABC-MCMC algorithm is implemented in the same MATLAB environment (version 9.3, 64-bit) as before with Intel® Core™ i5-6500 CPU @3.20 GHz processor and 8.00 GB RAM memory.

Prior and Proposal Distributions

To estimate all six gamma process parameters using ABC-MCMC, the following uninformative prior distributions are chosen: $f(\alpha) = \mathcal{U}[0, 50]$, $f(\eta) = \mathcal{U}[0, 5]$, $f(\beta) = \mathcal{U}[0, 1]$, $f(\mu_A) = \mathcal{U}[0, 5]$, $f(\sigma_A) = \mathcal{U}[0, 1]$, and $f(\sigma_Z) = \mathcal{U}[0, 1]$. To reduce the burn-in time in ABC-MCMC, the initialization scheme presented in Algorithm 13 is employed. The COV of the initialization proposal distribution is considered twice the COV of the proposal distribution used in the ABC-MCMC algorithm.

Distance Function and Tolerance Threshold

For ABC-MCMC, the following distance function is used:

$$\rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}}) = \|(d_1, d_2, \cdots, d_N)\| \tag{5.4}$$

where $d_i = \|\Delta\mathbf{y}_i^* - \Delta\mathbf{y}_i\|$. For the choice of the tolerance threshold, a similar method suggested by Beaumont et al. [12] is adopted in this study. However, the authors proposed the method in an ABC rejection sampler setting, whereas here we apply the same idea in the ABC-MCMC environment. Accordingly, ϵ is set to be a percentile Q_ϵ of the distribution of the proposed distance function values obtained from the proposed moves (samples) of the corresponding MCMC runs. However, the distribution of $\rho(\mathbf{D}, \mathbf{D}_{\text{obs}})$ is generally not available a priori; thus, the algorithm can be initialized with a chosen ϵ and later a check can be performed by evaluating whether the chosen ϵ follows the criterion $\mathbb{P}[\rho(\mathbf{D}, \mathbf{D}_{\text{obs}}) \leq \epsilon] \leq Q_\epsilon/100$. In this study, a value of $Q_\epsilon = 0.01$ is selected for the choice of tolerance threshold, which means that the chosen ϵ value should be less than the 0.01th percentile of the distribution of $\rho(\mathbf{D}, \mathbf{D}_{\text{obs}})$ obtained from the proposed moves by the proposal distribution.

Burn-In, Thinning, and Convergence

The "burn-in" and "thinning" of the Markov chains (for more details, see Section 3.4) can be equivalently applied to the ABC-MCMC method. To monitor the convergence of ABC-MCMC, the approach based on the GR statistic (Section 3.4.3) is employed. In this study, we calculate the GR statistic $\hat{R}$ using three MCMC runs for each parameter, and the samples from all three chains are used to make further statistical inferences. A cutoff criterion of $\hat{R} \leq 1.01$ is used to determine the iteration length. A flowchart describing the steps of the proposed scheme is presented in Figure 5.13.

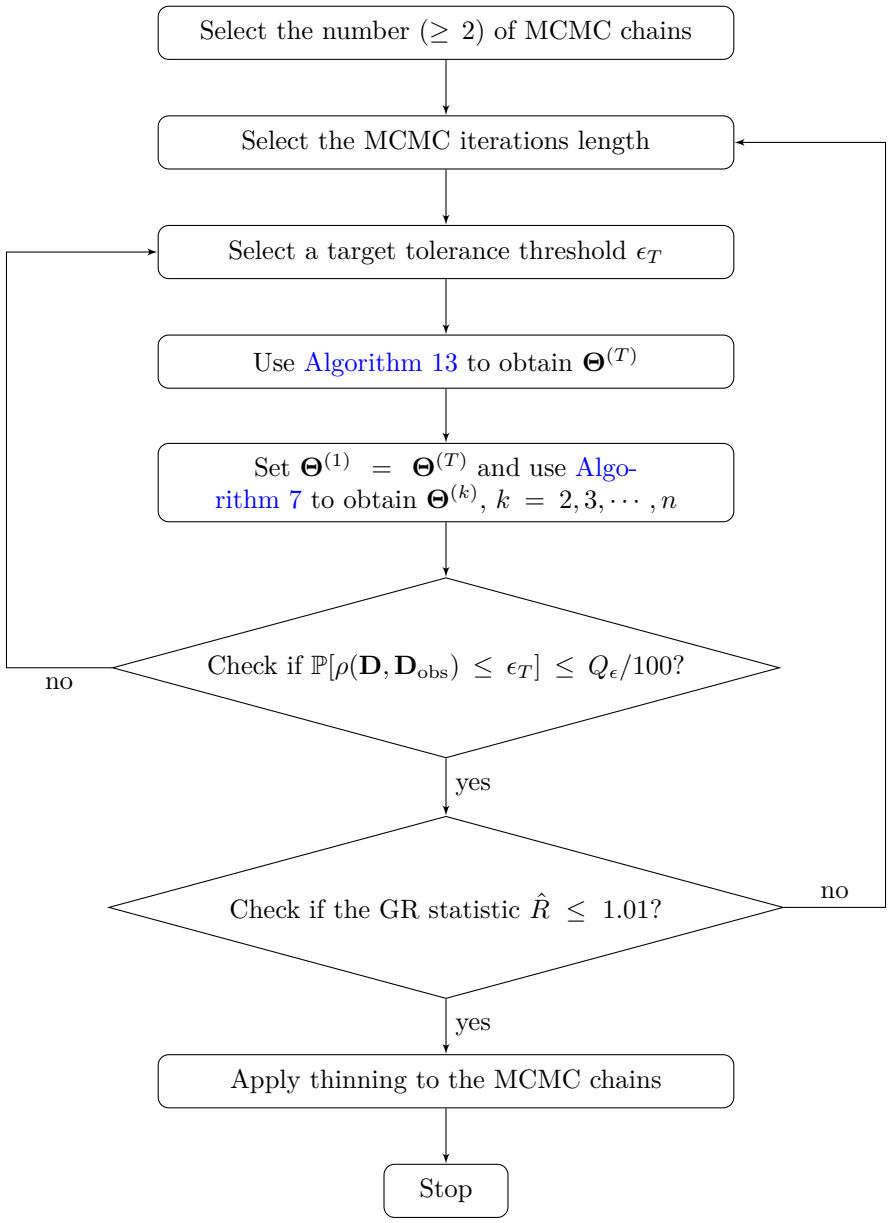


Figure 5.13: Flowchart of the proposed ABC-MCMC scheme.

5.3.3 Results and Discussion

Parameter Estimates

In an attempt to improve the convergence and acceptance rate of the ABC-MCMC scheme, the model parameters are updated one-by-one sequentially, which avoids very long sample chains. The marginal posterior distributions of the model parameters are shown in Figure 5.14. The means, COVs, and 95% CIs of the parameters are presented in Table 5.4. The mean of the time parameter η is found to be around 1.6, which proves that the underlying degradation process is mostly non-stationary. Except the mean initial degradation parameter μ_A and the time parameter η, all other parameters show very high uncertainties reflected by their COV values. It can be observed in the figure that the standard deviation parameter of the measurement noise σ_Z is quite small. This indicates that the estimation of such parameters are difficult, which is also reported in the simulation example in Section D.3.

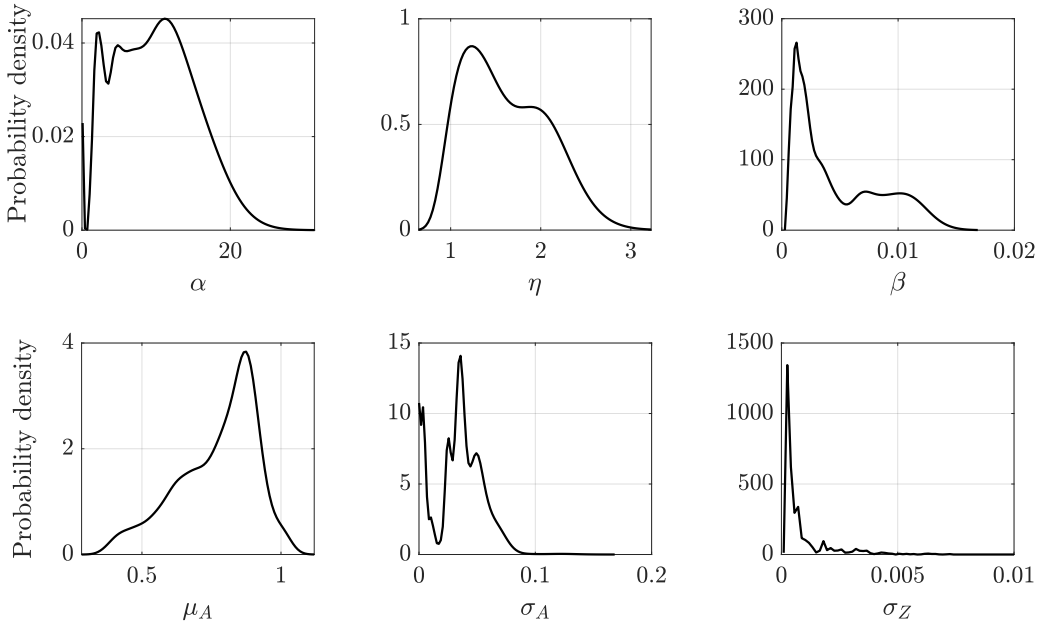


Figure 5.14: Marginal posterior distributions of the gamma process parameters.

Table 5.4: Statistical properties of the estimated parameters.

Parameter	Mean	COV	95% CI
α	6.7161	0.9155	[0.1358, 19.012]
η	1.5985	0.2504	[1.0122, 2.1135]
β	0.0048	0.7678	[0.0007, 0.0118]
μ_A	0.7715	0.1841	[0.4375, 0.9902]
σ_A	0.0153	1.4075	[0.0001, 0.0642]
σ_Z	0.0007	1.7409	[0.0001, 0.0037]

Degradation Prediction and Lifetime Distribution

With the help of simulation, the posterior parameter sample sets are used for predicting the mean degradation path along with the 95% CI as shown in Figure 5.15. It can be observed that the degradation path increases over time and the uncertainty in prediction also increases once it passes the observation data points, which is quite obvious. The figure shows that the upper bound of the predicted degradation growth crosses the nominal thickness at around 42 EFPY.

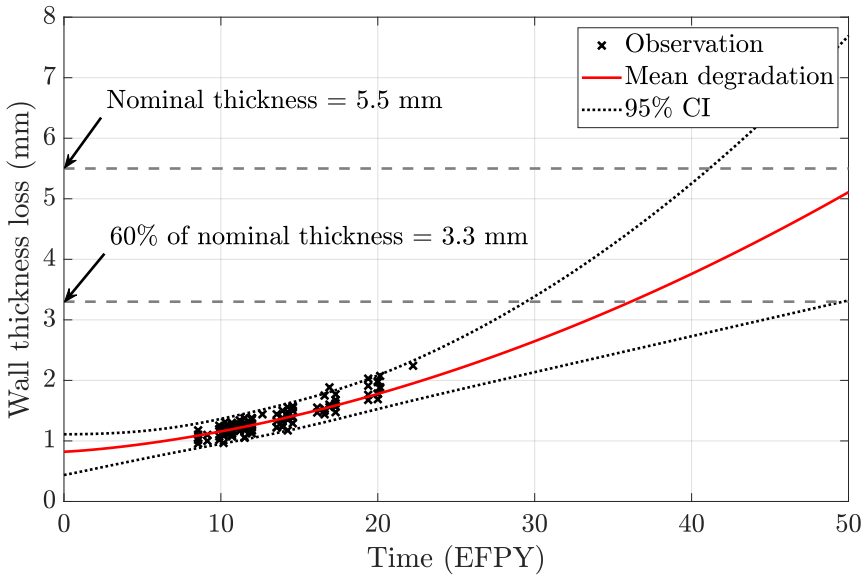


Figure 5.15: Predicted mean degradation growth with 95% credible interval (CI).

Assuming a wall thickness loss of 60% of the nominal thickness to be a critical limit, the distributions of mean and fifth percentile of the population lifetime (often a quantity of interest for maintenance planning) are plotted in Figure 5.16. The statistical properties of the lifetime quantiles are presented in Table 5.5. According to the table, one can expect the feeder pipes to reach their end of life by around 37.39 EFPY. Another quantity of interest is the fifth percentile of fifth percentile lifetime $T_{05/05}$, which is found to be around 28.74 EFPY. The lifetime distributions show that both mean and fifth percentile lifetimes have high uncertainties, which may result from limited observations; this justifies the application of the Bayesian method. Overall, given only a small data set, it is found that ABC-MCMC performed reasonably well in estimating all six parameters of the gamma process.

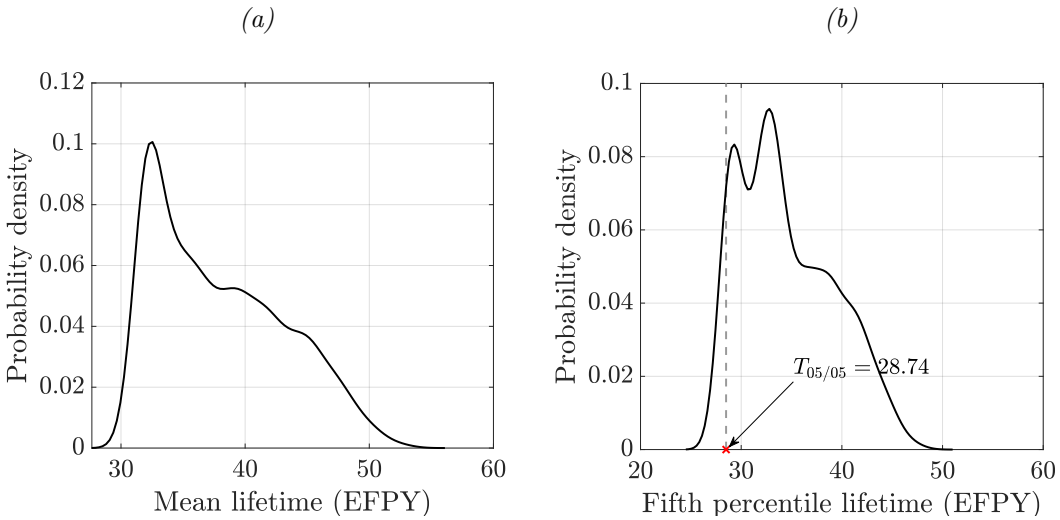


Figure 5.16: (a) Mean and (b) fifth percentile of the lifetime distribution.

Table 5.5: Statistical properties of the lifetime distribution.

Lifetime	Mean (EFPY)	COV	95% CI
Mean	37.39	0.1381	[31.36, 48.73]
Fifth percentile	33.57	0.1351	[27.61, 44.15]

5.4 Example III: Mixed-Effects Regression Model for Degradation Data

As an alternative to the random rate model, a better choice to model the FAC process is the more advanced LMER model (Subsection 2.2.3). The reason behind this is that the LMER model can be used to obtain degradation characteristics of individual components since it considers both system-level fixed effects as well as component-level random effects. The Bayesian inference method is used to estimate the regression parameters when the degradation data are limited, and at the same time, confounded by measurement uncertainties. The Gibbs sampler, commonly used for this purpose, works when the regression errors are assumed normally distributed which allows for the analytical formulation of the likelihood function. In case of a more general regression error distribution (e.g., mixture models), the likelihood becomes analytically intractable and computationally expensive to a degree that any likelihood-based Bayesian inference scheme can no longer be used as a practical method.

In this example, the application of the ABC method is extended to the LMER model which is essentially a two-stage hierarchical model. The sizing and coverage error issues associated with degradation measurements have been taken into account. The ABC-SS method (Algorithm 8) is implemented to estimate the parameters of the LMER model. Since the LMER model has too many parameters to be estimated, the sequential ABC-SS method is chosen over the ABC-MCMC method. Moreover, numerical investigations are carried out to evaluate and compare the performance of the ABC-SS algorithm under different distance settings.

5.4.1 Degradation Data and Model

Data Set

Minimum wall thickness data from a total of 62 feeder pipes are recorded from a station. These pipes have a nominal thickness of 5.5 mm and a common diameter of 2 inches. These feeder pipes are inspected repeatedly, where the number of repeat measurements ranges from one to five times. To be more specific, out of 62 feeder pipes, 19 pipes are inspected once, 19 inspected twice, 12 inspected thrice, 3 inspected four times, and the rest, 9, are inspected five times each. The inspection time varies between 8.55 and 22.25 EFPY.

Model

The degradation data is modeled using the LMER model. For its basic properties and the derivation of the likelihood function, the reader is referred to Subsection 2.2.3. The measurement error is modeled using a flexible mixture of distributions model; see the next subsection for more details.

5.4.2 Implementation Details

The ABC-SS algorithm is implemented in the MATLAB environment (version 9.3, 64-bit) with Intel® Core™ i5-6500 CPU @3.20 GHz Processor and 8.00 GB memory (RAM). The convergence of the ABC-SS algorithm is assessed by running the algorithm ten times. The posterior samples obtained from all ten runs of the algorithm are used to make further statistical inferences.

Distance Function

In this study, we introduced ten potential candidates for the ABC-SS distance function that can be used for degradation modeling. Our choices of distance functions take into consideration the absence of any specific rule to choose a particular distance function as well as our interest in comparing how the other types of distances perform. The selected distance functions are averaged over the number of components N to accommodate multiple repeated measurements. The distance functions are as follows (for more details, see [40]):

- Manhattan (ℓ_1) distance:

$$\rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}}) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{y}_i^{(k)} - \mathbf{y}_i\right\|_1 = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m_i}\left|y_{ij}^{(k)} - y_{ij}\right| \tag{5.5}$$

- Euclidean (ℓ_2) distance:

$$\rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}}) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{y}_i^{(k)} - \mathbf{y}_i\right\|_2 = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=1}^{m_i}\left|y_{ij}^{(k)} - y_{ij}\right|^2\right)^{\frac{1}{2}} \tag{5.6}$$

- Normalized Manhattan (N-ℓ_1) distance:

$$\rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\left\|\mathbf{y}_i^{(k)} - \mathbf{y}_i\right\|_1}{\left\|\mathbf{y}_i^{(k)}\right\|_1 + \left\|\mathbf{y}_i\right\|_1} \tag{5.7}$$

- Normalized Euclidean (N-ℓ_2) distance:

$$\rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\left\|\mathbf{y}_i^{(k)} - \mathbf{y}_i\right\|_2}{\left\|\mathbf{y}_i^{(k)}\right\|_2 + \left\|\mathbf{y}_i\right\|_2} \tag{5.8}$$

- Chebyshev (ℓ_∞) distance:

$$\rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}}) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{y}_i^{(k)} - \mathbf{y}_i\right\|_\infty = \frac{1}{N}\sum_{i=1}^{N}\max_j\left(\left|y_{ij}^{(k)} - y_{ij}\right|\right) \tag{5.9}$$

- Soergel (Soer) distance:

$$\rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{j=1}^{m_i}\left|y_{ij}^{(k)} - y_{ij}\right|}{\sum_{j=1}^{m_i}\max\left\{y_{ij}^{(k)}, y_{ij}\right\}} \tag{5.10}$$

- Symmetric χ^2 (S-χ^2) measure:

$$\rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m_i}\frac{2}{\overline{y}_i^{(k)}\cdot\overline{y}_i}\cdot\frac{\left(y_{ij}^{(k)}\overline{y}_i - y_{ij}\overline{y}_i^{(k)}\right)^2}{y_{ij}^{(k)} + y_{ij}} \tag{5.11}$$

- Sørensen (Sør) distance:

$$\rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{j=1}^{m_i}\left|y_{ij}^{(k)} - y_{ij}\right|}{\sum_{j=1}^{m_i}\left(y_{ij}^{(k)} + y_{ij}\right)} \tag{5.12}$$

- Canberra (Can) distance:

$$\rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m_i}\frac{\left|y_{ij}^{(k)} - y_{ij}\right|}{\left|y_{ij}^{(k)}\right| + |y_{ij}|} \tag{5.13}$$

- Kulczynski (Kul) distance:

$$\rho(\mathbf{D}^{(k)}, \mathbf{D}_{\text{obs}}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{j=1}^{m_i}\left|y_{ij}^{(k)} - y_{ij}\right|}{\sum_{j=1}^{m_i}\min\left\{y_{ij}^{(k)}, y_{ij}\right\}} \tag{5.14}$$

Along with Manhattan and Euclidean distances, we selected their normalized versions to investigate the impact of normalization. Among these ten distances, four of them,

Soergel, Symmetric χ^2, Sørensen, and Kulczynski, may produce negative distance values, although that would be rare since the degradation measurements are generally non-negative. However, in such a situation, the absolute value of the distance function is considered.

Choice of Acceptance Probability and other Algorithmic Hyperparameters

The accuracy of the proposed ABC-SS algorithm is controlled by the final tolerance threshold $\epsilon^{(S)}$ which needs to be very small. The tolerance threshold $\epsilon^{(s)}$ of each simulation level is chosen adaptively with the help of the acceptance probability p_0. A too small p_0 value may produce unsatisfactory results due to a high number of sample repetitions. In contrast, a too high p_0 value may end up generating substandard posterior samples that belong to high tolerance values. The final tolerance threshold $\epsilon^{(S)}$ is directly dependent on the acceptance probability of the ABC posterior samples at the Sth simulation level, i.e., the final acceptance probability p_0^S. Thus, to attain a very small $\epsilon^{(S)}$ value, careful selection of the hyperparameters p_0 and S is of paramount importance. In practice, one may select the values of p_0 and S by monitoring the tolerance threshold $\epsilon^{(s)}$ of each simulation level. The algorithm can be stopped when the change in the final tolerance $\epsilon^{(S)}$ is minimal compared to the tolerance $\epsilon^{(S-1)}$ of the previous simulation level. To guarantee the convergence of the tolerance level, we chose to stop the algorithm when only 5% change is observed in the final tolerance threshold. Our investigation revealed that a tolerance change of less than 5% induces high sample repetition thus should be avoided. On the other hand, the maximum number of simulation levels allowed per run are set at 30 irrespective of the tolerance criterion. Note that the value of p_0 determines the number of simulation levels required. A large p_0 value will need large number of simulation levels to get satisfactory results, whereas a small p_0 value will help to quickly converge to a small final tolerance requiring

less number of simulation levels. Although, Chiachio et al. [30] suggested a value of 0.2 for p_0, in this study, we selected three values for p_0, 0.1, 0.2, and 0.25, to find the best performing one (in terms of quick convergence and sample repetitions) under the current settings of the degradation model and different distance measures.

The number of model simulations n_0 at each simulation level determines the computational cost of the algorithm. The total number of model simulations $n_0 S$ should be optimized to achieve high computational efficiency. Chiachio et al. [30] used different n_0 values, 1000 and 2000, in the examples that consist of a moving average process of order two and a single DOF linear oscillator subjected to white noise excitation. However, since the degradation model has a high-dimensional parameter space, we selected a higher value of 10000 for the parameter n_0. The advantage of setting a high value for n_0 is that it helps to reduce sample repetitions generating better posterior samples, and sometimes quick convergence to a much smaller threshold value which otherwise can not be achieved.

Inspection Error

The wall thickness data give information about the extent of degradation due to FAC on the inner walls of the feeder pipes. In nuclear power plants, these wall thicknesses are measured manually using bracelet type ultrasonic tools by mounting them on the feeder pipes [94]. Figure 5.17 shows such a typical bracelet type ultrasonic tool (known as "14-probe" scanner). These electronic tools often contaminate the data by adding random noise or sizing error to the measurements [60]. The sizing error is generally modeled as a normally distributed random variable with zero mean and unknown standard deviation [60, 61, 79]. Jyrkama and Pandey [70] reported that, in addition to the sizing error, the wall thickness data are contaminated by coverage error. The coverage error appears when wall thickness measurements are taken at certain points

instead of measuring the entire surface of the piping components. Intrinsic to the feeder inspection tools, coverage error occurs for two main reasons: (1) manual placement and operation of the probes, and (2) fixed spacing between individual transducers of the tool. As a result, the minimum thickness of a pipe reported from the inspection is either equal to or more than the true minimum thickness of a piping component. Thus, the coverage error can be modeled as a random variable that has a positive support. Investigations by Jyrkama and Pandey [70] reveal that the coverage error distribution is slightly skewed to the right.

Figure 5.17: A typical bracelet type ultrasonic tool used for feeder pipe inspections. (Image courtesy of ZETEC®, reproduced with permission.)

In this paper, we select a mixture of distributions to model the error term z_{ij} in Equation 2.17. The mixture model $f(z)$ consists of the normal distribution $\mathcal{N}(0, \sigma_z^2)$ and the gamma distribution $\mathcal{G}(\alpha_z, \beta_z)$ each having equal weights; σ_z is the standard deviation of the sizing error, and α_z and β_z are the shape and scale parameters of the

coverage error, respectively. Thus, the distribution of the error can be written as

$$f(z) = 0.5\,\mathcal{N}(0, \sigma_z^2) + 0.5\,\mathcal{G}(\alpha_z, \beta_z) \tag{5.15}$$

The parameters of the mixture model are assumed to be unknown. Details about the specification of the prior distributions for these parameters are presented in the next subsection.

Prior Distributions

Let us denote the model parameters as $\boldsymbol{\Theta} = \{\boldsymbol{\beta}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_N, \boldsymbol{\Sigma}_b, \tau\}$, where $\boldsymbol{\beta}_i = \boldsymbol{\beta} + \mathbf{b}_i$, and $\tau = \{\sigma_z, \alpha_z, \beta_z\}$ is the set of distribution parameters of the error term. Let us assume that β_0 represents the system-level initial pipe wall thickness, and β_1 represents the system-level wall thinning rate due to FAC (or simply FAC rate), whereas b_{0i} and b_{1i} represent the corresponding variations. The prior distributions of the individual parameters are assumed to be independent.

Sometimes, the initial wall thicknesses of feeder pipes are not known precisely because they vary from their nominal value due to bending or welding operations. Recent investigations [60, 61] show that the variations are significant, in fact, the initial wall thicknesses get reduced to as low as around 80% of the nominal thickness of the pipes. With that information at hand, the prior distribution for β_0 is selected to be a normal distribution with a mean equal to the nominal thickness of 5.5 mm, and a standard deviation of 1 mm, i.e., $f(\beta_0) = \mathcal{N}(5.5, 1)$. On the other hand, a prior of $f(\beta_1) = \mathcal{N}(0, 0.1^2)$ is chosen for the FAC rate based on the fact that FAC rates in the feeder pipes generally turn out to be of the order of 10^{-2} mm/EFPY [60, 94, 98].

The covariance matrix $\boldsymbol{\Sigma}_b$ consists of three parameters: σ_0, σ_1, and ρ. To specify the prior distributions for these three parameters, knowledge about the variation of the initial wall thickness and FAC rate is needed. Investigations by Hazra et al. [60, 61]

reveal that the variation in initial wall thickness of feeder pipes may reach up to 20% of the nominal thickness, whereas the variation of the FAC rate is well within around 10% of the system-level rate. Based on this information, we select the following priors for the above-mentioned parameters: $f(\sigma_0) = f(\sigma_1) = \mathcal{G}(4, 0.1)$ and $f(\rho) = \mathcal{N}(0, 0.2^2)$.

According to the LMER model assumptions, we can write $f(\boldsymbol{\beta}_i \mid \boldsymbol{\beta}, \boldsymbol{\Sigma}_b) = \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{\Sigma}_b)$ which is a bivariate normal density function. After generating $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}_b$ from their respective priors, one can simply sample from this bivariate normal density to generate prior samples of $\boldsymbol{\beta}_i$ for all $i = 1, 2, \cdots, N$.

The standard deviation of the sizing error is generally found to be quite small (within the order of 10^{-1} mm) compared to the wall thickness measurements of feeder pipes [60, 79]. Therefore, we select a gamma prior $f(\sigma_z) = \mathcal{G}(4, 0.1)$ for the standard deviation parameter σ_z. On the other hand, the study by Jyrkama and Pandey [70] shows that the properties of the coverage error change depending on the type of inspection tool. However, the mean and standard deviation of the coverage error are found to be in the order of 10^{-2} mm. Using this information, we selected the following prior distributions for the coverage error parameters: $f(\alpha_z) = \mathcal{U}[0, 10]$ and $f(\beta_z) = \mathcal{U}[0, 0.1]$, where $\mathcal{U}[\bullet, \bullet]$ represents the uniform density.

Data Simulation

An important step in the ABC-SS algorithm is forward simulation of the underlying model that generates a pseudo aggregate data set $\mathbf{D}$. The simulated data $\mathbf{D}$ are used to calculate the distance $\rho(\mathbf{D}, \mathbf{D}_{\text{obs}})$ between $\mathbf{D}$ and the observed data $\mathbf{D}_{\text{obs}}$. However, to get the best results, the process of data simulation should mimic exactly the data generation process of the observed data $\mathbf{D}_{\text{obs}}$ [27]. Once the prior distributions are selected, samples of the set of model parameters $\boldsymbol{\Theta}$ are generated from the respective priors. Using each sample of $\{\boldsymbol{\beta}, \boldsymbol{\Sigma}_b\}$, one can generate the pipe-specific parameters

$\{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_N\}$, where $N = 62$, from the bivariate normal density function $\mathcal{N}(\boldsymbol{\beta}, \boldsymbol{\Sigma}_b)$. These pipe-specific parameters can be used to generate true degradation growths $\mathbf{T}_i\boldsymbol{\beta}_i$ for each of the ith pipe components. Next, the error terms are simulated from the density function $f(z)$, the expression of which is given in Equation 5.15. Each error term is simulated from $\mathcal{N}(0, \sigma_z^2)$ with 50% probability and from $\mathcal{G}(\alpha_z, \beta_z)$ with 50% probability. Afterwards, the error vectors $\mathbf{z}_i$, $i = 1, 2, \cdots, N$, are generated and added to the true degradation growths as $\mathbf{y}_i = \mathbf{T}_i\boldsymbol{\beta}_i + \mathbf{z}_i$, for $i = 1, 2, \cdots, N$, to simulate a complete pseudo aggregate data set $\mathbf{D}$.

5.4.3 Results and Discussion

Parameter Estimates

The posterior estimates of the model parameters in terms of their means and 95% CIs are shown in Figure 5.18. As shown by Figure 5.18a and Figure 5.18b, the system-level parameters β_0 and β_1 are successfully inferred by ABC-SS under all the distance functions except the Symmetric-χ^2 measure which completely fails to capture the uncertainty of the parameter β_0. Although $p_0 = 0.1$ can be seen producing tighter parameter estimates, the Canberra distance appears to have less effect on the selection of the acceptance probability. Figure 5.18c, Figure 5.18d, and Figure 5.18e show the estimates of the covariance matrix parameters. The symmetric-χ^2 measure produces an unusual estimate for the parameter σ_0. The effect of p_0 is visible in all the estimates of the covariance parameters, which implies that a smaller p_0 is a better choice. Performances of other distance functions except the symmetric-χ^2 measure are very similar. The estimates of the noise parameters σ_z, a_z, and b_z are shown in Figure 5.18f, Figure 5.18g, and Figure 5.18h, respectively. Once again, the symmetric-χ^2 measure failed to capture the uncertainty of the parameter σ_z accurately. While the effects of

distance functions and p_0 values are evident on the estimates of σ_z, they seem to show no significant effect on the estimates of the other noise parameters, namely a_z and b_z. The posteriors produced in Figure 5.18 using different distance functions and different values of p_0 show large differences between the parameter ranges. If we focus only on the effect of the p_0 values, it is clear from the figure that $p_0 = 0.1$ produces the best parameter estimates since it produces tighter intervals of the model parameters. The acceptance probabilities $p_0 = 0.2$ and 0.25 do not seem to produce noticeable difference among the parameter estimates. Comparing the distance functions, Figure 5.18a, Figure 5.18c, and Figure 5.18f clearly show that the performance of the symmetric-χ^2 measure is the worst particularly for estimating the parameters β_0, σ_0, and σ_z.

Table 5.6 presents the numerical values of means and COVs of the parameters along with the number of simulation levels reached after convergence, the final tolerances, and computation times. As seen, in all cases, the numbers of simulation levels reached after satisfying the stopping criterion of 5% tolerance difference are well below 30 which is the maximum allowable number of simulation levels. Notice that the numbers of simulation levels reached after convergence are far less for $p_0 = 0.1$ than the other values of p_0. Moreover, for $p_0 = 0.1$, the final tolerances achieved smaller values compared to $p_0 = 0.2$ and 0.25. This confirms that a smaller acceptance probability p_0 in ABC-SS generates posterior samples with higher accuracy (since it quickly achieves smaller final tolerance) and less computation time in general (since it requires fewer simulation levels). For almost all cases, with $n_0 = 10000$ samples, the computation time for 10 runs of the ABC-SS algorithm is found to be around half an hour. The means of the system-level parameters β_0 and β_1 are found to be around 5.1 mm and -0.07 mm/EFPY, respectively. The uncertainties of the parameters are captured through the COVs. The smallest COVs of these two parameters are given by Manhattan (ℓ_1), Euclidean (ℓ_2), normalized-Euclidean (N-ℓ_2), and Chebyshev (ℓ_∞) distance functions for $p_0 = 0.1$ case,

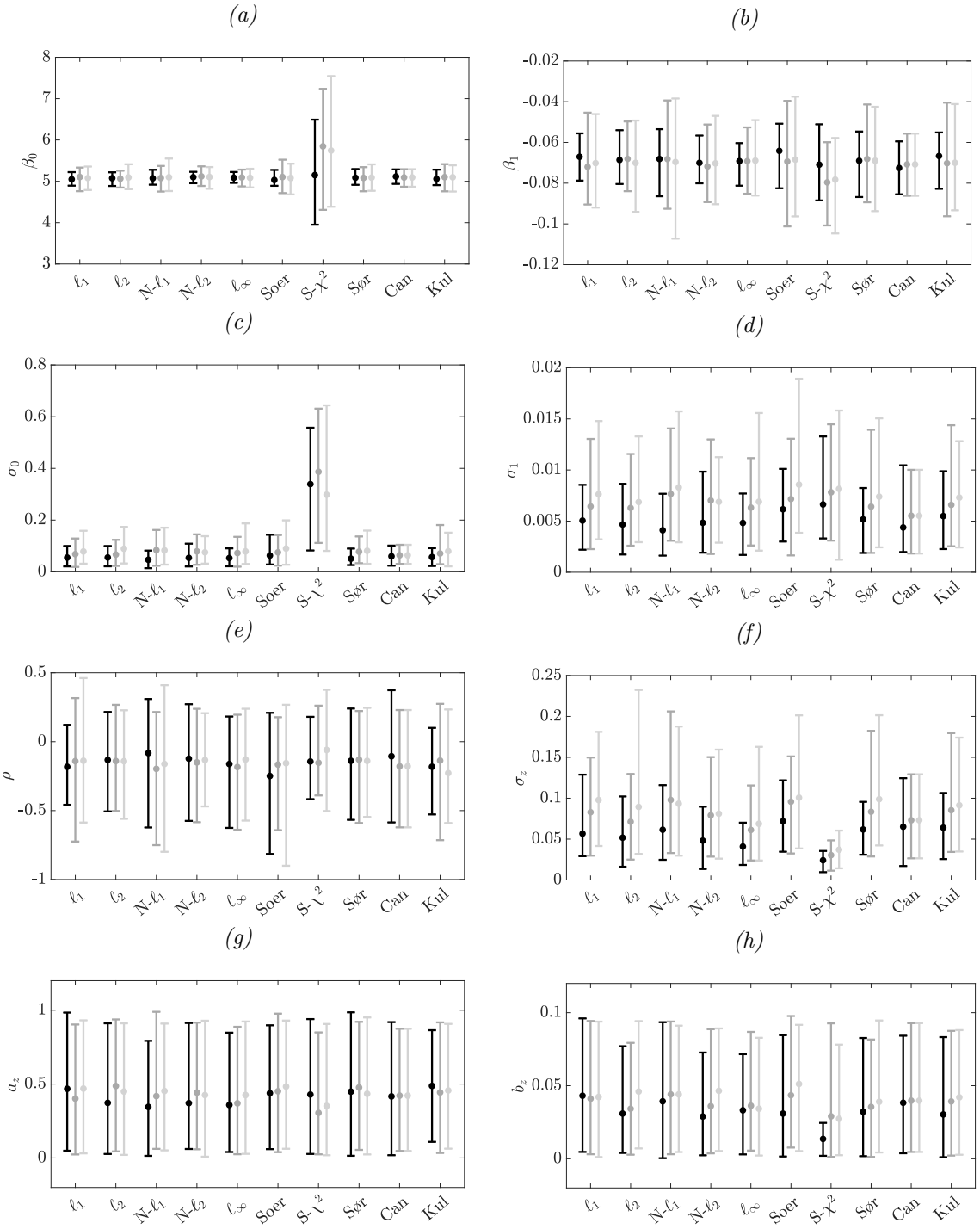


Figure 5.18: Mean values (filled circles) and 95% credible bounds (error bars) of the posterior samples of model parameters. Results generated using $p_0 = 0.1$ are in black, $p_0 = 0.2$ in dark gray, and $p_0 = 0.25$ in light gray.

although the other distance functions, except the symmetric χ^2, produce reasonably accurate estimates. The COV values for β_0 and β_1 obtained using $p_0 = 0.1$ are found to be around 1.5% and 8.5%, respectively, meaning that the system-level FAC rate β_1 has more uncertainty than the system-level initial wall thickness parameter β_0. The symmetric χ^2 measure performs poorly compared to other distances, hence, its use should be avoided in the ABC-SS algorithm. The means of the standard deviation parameters σ_0 and σ_1 are found to be in the order of 10^{-2} and 10^{-3}, respectively. This implies that the variations of the individual pipe-specific parameters are quite small. Nevertheless, these two parameters seem to have very high uncertainties as depicted by their respective COVs that are around 30%. The mean of the correlation coefficient parameter ρ is found to be around -0.2, although the high COV values obtained using different distance functions indicate very high uncertainty in the parameter. Surprisingly, the Soergel, Sørensen, and Kulczynski distance functions performed quite well in estimating the parameters of the covariance matrix $\mathbf{\Sigma}_b$. The noise parameter σ_z is found to be in the order of 10^{-2}. Soergel and Sørensen distance functions performed quite well in estimating the COV of σ_z, which is found to be around 30%. The other two noise parameters a_z and b_z seem to have very high uncertainties, and less effect on the selection of the distance function. A plausible reason could be that, compared to other model parameters, these two noise parameters suffer from unidentifiability problems. It can be noted that Manhattan (ℓ_1), Euclidean (ℓ_1), normalized-Euclidean (N-ℓ_2), and Chebyshev (ℓ_∞) distance functions worked quite well in estimating the fixed effects parameters, whereas the Soergel, Sørensen, and Kulczynski distance functions are found to be well suited for estimating the standard deviation parameters.

Table 5.6: Means and COVs (in brackets) of the posterior distributions, the number of simulation levels, final tolerances, and computation times of different runs of the ABC-SS algorithm.

Distance function	p_0	β_0	β_1	σ_0	σ_1	ρ	σ_z	a_z	b_z	S	$\epsilon^{(S)}$	Computation time
	0.1	**5.049(0.015)**	**−0.067(0.086)**	0.055 (0.415)	0.005 (0.396)	**−0.182(1.043)**	0.057 (0.429)	0.468 (0.826)	0.043 (0.617)	$14 \sim 19$	$0.212 \sim 0.263$	29 min 44 sec
ℓ_1	0.2	5.103 (0.027)	−0.072(0.150)	0.068 (0.405)	0.006 (0.416)	−0.141(1.708)	0.083 (0.362)	0.401 (0.687)	0.041 (0.666)	$15 \sim 20$	$0.269 \sim 0.364$	31 min 10 sec
	0.25	5.078 (0.029)	−0.070(0.165)	0.079 (0.420)	0.008 (0.390)	−0.138(1.603)	0.098 (0.375)	0.469 (0.607)	0.042 (0.669)	$16 \sim 22$	$0.288 \sim 0.445$	31 min 38 sec
	0.1	**5.074(0.015)**	**−0.069(0.088)**	**0.055(0.358)**	0.005(0.379)	−0.132(1.576)	0.052 (0.467)	0.373 (0.753)	0.031 (0.687)	$14 \sim 17$	$0.146 \sim 0.184$	28 min 47 sec
ℓ_2	0.2	5.064 (0.021)	−0.068(0.127)	0.067 (0.394)	0.006(0.367)	−0.140(1.426)	0.071 (0.392)	0.487 (0.621)	0.034 (0.682)	$15 \sim 21$	$0.178 \sim 0.252$	31 min 40 sec
	0.25	5.093 (0.029)	−0.070(0.165)	0.089 (0.401)	0.007 (0.387)	−0.141(1.425)	0.089 (0.452)	0.449 (0.622)	0.046 (0.602)	$14 \sim 21$	$0.198 \sim 0.324$	30 min 30 sec
	0.1	5.071 (0.021)	−0.068(0.134)	**0.047(0.364)**	0.004 (0.413)	−0.083(2.517)	0.061 (0.380)	0.345 (0.718)	0.039 (0.770)	$14 \sim 17$	$0.012 \sim 0.014$	30 min 24 sec
N-ℓ_1	0.2	5.078 (0.034)	−0.068(0.201)	0.084 (0.409)	0.008 (0.394)	−0.197(1.240)	0.098 (0.488)	0.418 (0.709)	0.044 (0.678)	$14 \sim 20$	$0.014 \sim 0.021$	29 min 32 sec
	0.25	5.097 (0.039)	−0.070(0.242)	0.083 (0.458)	0.008 (0.404)	−0.162(1.769)	0.093 (0.428)	0.453 (0.598)	0.044 (0.643)	$17 \sim 21$	$0.015 \sim 0.020$	31 min 46 sec
	0.1	**5.099(0.014)**	**−0.070(0.082)**	0.054 (0.412)	0.005 (0.487)	−0.123(1.867)	0.048 (0.440)	0.371 (0.845)	0.029 (0.837)	$15 \sim 16$	$0.013 \sim 0.014$	36 min 08 sec
N-ℓ_2	0.2	5.119 (0.027)	−0.072(0.154)	0.079 (0.387)	0.007 (0.495)	−0.150(1.417)	0.079 (0.401)	0.443 (0.642)	0.036 (0.701)	$16 \sim 21$	$0.014 \sim 0.018$	36 min 43 sec
	0.25	5.104 (0.025)	−0.070(0.149)	**0.075(0.366)**	0.007(0.333)	−0.133(1.318)	0.081 (0.400)	0.425 (0.716)	0.047 (0.594)	$17 \sim 22$	$0.016 \sim 0.021$	39 min 25 sec
	0.1	**5.086(0.015)**	**−0.069(0.087)**	0.053 (0.390)	0.005(0.339)	−0.162(1.406)	0.041 (0.351)	0.358 (0.707)	0.033 (0.666)	$15 \sim 18$	$0.121 \sim 0.140$	35 min 16 sec
ℓ_∞	0.2	5.093 (0.019)	−0.069(0.112)	0.073 (0.383)	0.006 (0.389)	−0.184(1.179)	0.061 (0.418)	0.370 (0.734)	0.036 (0.730)	$16 \sim 20$	$0.151 \sim 0.182$	36 min 32 sec
	0.25	5.083 (0.024)	−0.069(0.143)	0.079 (0.494)	0.007 (0.475)	−0.129(1.547)	0.069 (0.491)	0.426 (0.657)	0.034 (0.731)	$13 \sim 24$	$0.141 \sim 0.295$	39 min 25 sec
	0.1	5.032 (0.021)	−0.064(0.136)	0.063 (0.488)	**0.006(0.307)**	**−0.249(1.063)**	**0.072(0.338)**	0.438 (0.636)	0.031 (0.850)	$13 \sim 17$	$0.024 \sim 0.030$	33 min 33 sec
Soer	0.2	5.102 (0.035)	−0.069(0.204)	0.075 (0.426)	0.007 (0.401)	−0.166(1.344)	0.096 (0.326)	0.451 (0.689)	0.044 (0.641)	$16 \sim 22$	$0.025 \sim 0.035$	36 min 01 sec
	0.25	5.079 (0.035)	−0.068(0.221)	0.090 (0.573)	0.009 (0.493)	−0.156(1.511)	0.101 (0.402)	0.483 (0.562)	0.051 (0.550)	$16 \sim 21$	$0.028 \sim 0.039$	35 min 29 sec
	0.1	5.150 (0.138)	−0.071(0.143)	0.339 (0.426)	0.007 (0.460)	−0.144(1.241)	0.024(0.293)	0.429 (0.598)	0.014 (0.904)	$13 \sim 17$	$0.0008 \sim 0.0011$	40 min 02 sec
S-χ^2	0.2	5.846 (0.130)	−0.080(0.135)	0.387 (0.390)	0.008 (0.380)	−0.153(1.216)	0.030 (0.375)	0.305 (0.945)	0.029 (0.893)	$17 \sim 20$	$0.0009 \sim 0.0012$	44 min 41 sec
	0.25	5.743 (0.129)	−0.078(0.136)	0.298 (0.469)	0.008 (0.475)	−0.060(3.471)	0.037 (0.367)	0.352 (0.872)	0.027 (0.822)	$17 \sim 22$	$0.0010 \sim 0.0015$	47 min 09 sec
	0.1	5.086 (0.020)	−0.069(0.121)	**0.058(0.339)**	**0.005(0.267)**	−0.139(1.525)	**0.062(0.305)**	0.449 (0.669)	0.032 (0.760)	$14 \sim 17$	$0.012 \sim 0.014$	34 min 44 sec
Sør	0.2	5.082 (0.030)	−0.068(0.184)	0.078 (0.374)	0.006 (0.414)	−0.131(1.476)	0.084 (0.428)	0.477 (0.582)	0.036 (0.732)	$16 \sim 19$	$0.014 \sim 0.018$	37 min 26 sec
	0.25	5.089 (0.034)	−0.069(0.211)	0.081 (0.430)	0.007 (0.425)	−0.139(1.570)	0.099 (0.445)	0.434 (0.702)	0.039 (0.725)	$14 \sim 22$	$0.014 \sim 0.026$	38 min 52 sec
	0.1	5.115 (0.019)	−0.073(0.103)	0.060 (0.384)	0.004 (0.532)	−0.106(2.647)	0.065 (0.444)	0.417 (0.752)	0.038 (0.632)	$13 \sim 17$	$0.027 \sim 0.036$	34 min 36 sec
Can	0.2	5.093 (0.022)	−0.071(0.117)	0.064 (0.419)	0.006 (0.431)	−0.179(1.171)	0.073 (0.367)	0.422 (0.605)	0.040 (0.713)	$17 \sim 21$	$0.030 \sim 0.038$	38 min 28 sec
	0.25	5.097 (0.032)	−0.071(0.176)	0.082 (0.404)	0.008 (0.410)	−0.136(1.452)	0.094 (0.414)	0.393 (0.731)	0.044 (0.641)	$16 \sim 23$	$0.031 \sim 0.049$	37 min 35 sec
	0.1	5.058 (0.020)	−0.067(0.122)	**0.057(0.367)**	0.006 (0.444)	**−0.182(1.014)**	0.064 (0.352)	**0.487(0.506)**	0.030 (0.824)	$14 \sim 17$	$0.024 \sim 0.028$	35 min 09 sec
Kul	0.2	5.099 (0.030)	−0.070(0.179)	0.071 (0.508)	0.007 (0.431)	−0.137(1.795)	0.085 (0.415)	0.443 (0.682)	0.039 (0.667)	$13 \sim 22$	$0.027 \sim 0.047$	38 min 12 sec
	0.25	5.098 (0.034)	−0.070(0.207)	0.079 (0.421)	0.007 (0.342)	−0.228(1.022)	0.091 (0.412)	0.457 (0.627)	0.042 (0.689)	$17 \sim 22$	$0.029 \sim 0.038$	38 min 30 sec

The posterior estimates of the corrosion parameters are presented in Table 5.7. The uncertainties of the parameters are represented using the respective COV values and 95% credible bounds. As seen, the mean of the initial wall thickness is found to be around 5.1 mm – a loss of around 7.3% of the nominal thickness, which indicates that the wall thickness losses are likely to occur at thinner sections such as extrados of the pipe bends. However, the variability of the initial pipe wall thickness seems to be much smaller compared to the FAC rate. The COV of the initial wall thickness is obtained to be around 2% – given by the best performing distance functions, Manhattan (ℓ_1), Euclidean (ℓ_1), normalized-Euclidean (N-ℓ_2), and Chebyshev (ℓ_∞), for the $p_0 = 0.1$ case. On the other hand, a mean of around 0.069 mm/EFPY and a COV of around 11% put the FAC rate in a position of high importance. Once again, the uncertainty of the FAC rate is best captured by the same four distance functions as above with the acceptance probability $p_0 = 0.1$. The upper bound FAC rate is found to be around 0.084 mm/EFPY. While the mean corrosion rate does not seem to be high, the upper bound rate of its 95% CI indicates a higher risk of pipe failure – eventually reducing the system lifetime. Better representations of the posterior samples and its variability obtained using $p_0 = 0.1$ are given by the estimates of the PDFs shown in Figure 5.19.

Lifetime Distribution and Survival Function

To calculate the system lifetime, 60% of the nominal thickness (2.2 mm) is chosen as a critical limit. The mean, COV, and 95% credible bounds of the lifetime distribution are presented in Table 5.8. Once again, the best lifetime estimates (given by smallest COVs and tighter intervals) are given by the following four distance functions: Manhattan (ℓ_1), Euclidean (ℓ_1), normalized-Euclidean (N-ℓ_2), and Chebyshev (ℓ_∞), for the $p_0 = 0.1$ case. The mean lifetime of the feeder pipe system is found to be around 42 EFPY, whereas its COV is found to be around 10% – a result of high uncertainty in the FAC

Table 5.7: Posterior estimates of the initial wall thickness and FAC rate of feeder pipes.

Distance fn.	p_0	Initial wall thickness (mm)			FAC rate (mm/EFPY)		
		Mean	COV	95% CI	Mean	COV	95% CI
ℓ_1	0.1	**5.049**	**0.019**	**[4.854, 5.248]**	**−0.067**	**0.119**	**[−0.082, −0.050]**
	0.2	5.104	0.030	[4.769, 5.383]	−0.072	0.178	[−0.096, −0.045]
	0.25	5.077	0.033	[4.742, 5.411]	−0.070	0.202	[−0.098, −0.042]
ℓ_2	0.1	**5.074**	**0.019**	**[4.874, 5.268]**	**−0.069**	**0.115**	**[−0.084, −0.052]**
	0.2	5.064	0.025	[4.807, 5.314]	−0.068	0.161	[−0.089, −0.046]
	0.25	5.092	0.035	[4.758, 5.465]	−0.070	0.196	[−0.098, −0.044]
N-ℓ_1	0.1	5.071	0.023	[4.871, 5.320]	−0.068	0.149	[−0.088, −0.051]
	0.2	5.078	0.038	[4.697, 5.447]	−0.068	0.234	[−0.098, −0.036]
	0.25	5.098	0.043	[4.668, 5.538]	−0.070	0.274	[−0.107, −0.031]
N-ℓ_2	0.1	**5.099**	**0.018**	**[4.901, 5.263]**	**−0.070**	**0.113**	**[−0.084, −0.053]**
	0.2	5.119	0.032	[4.817, 5.441]	−0.072	0.189	[−0.100, −0.047]
	0.25	5.104	0.030	[4.783, 5.388]	−0.070	0.181	[−0.095, −0.044]
ℓ_∞	0.1	**5.086**	**0.019**	**[4.915, 5.271]**	**−0.069**	**0.114**	**[−0.085, −0.054]**
	0.2	5.093	0.025	[4.841, 5.335]	−0.069	0.149	[−0.090, −0.049]
	0.25	5.083	0.030	[4.781, 5.364]	−0.069	0.181	[−0.092, −0.044]
Soer	0.1	5.032	0.025	[4.794, 5.296]	−0.064	0.169	[−0.086, −0.043]
	0.2	5.102	0.039	[4.714, 5.531]	−0.069	0.233	[−0.104, −0.038]
	0.25	5.079	0.041	[4.656, 5.471]	−0.068	0.261	[−0.102, −0.032]
S-χ^2	0.1	5.152	0.155	[3.690, 6.689]	−0.071	0.177	[−0.095, −0.046]
	0.2	5.845	0.148	[4.135, 7.652]	−0.079	0.171	[−0.108, −0.055]
	0.25	5.743	0.141	[4.269, 7.685]	−0.078	0.178	[−0.111, −0.054]
Sør	0.1	5.086	0.023	[4.873, 5.342]	−0.069	0.144	[−0.089, −0.050]
	0.2	5.082	0.034	[4.718, 5.410]	−0.068	0.211	[−0.095, −0.039]
	0.25	5.089	0.038	[4.697, 5.471]	−0.069	0.240	[−0.099, −0.036]
Can	0.1	5.115	0.023	[4.897, 5.337]	−0.073	0.123	[−0.089, −0.056]
	0.2	5.093	0.026	[4.832, 5.343]	−0.071	0.145	[−0.091, −0.051]
	0.25	5.097	0.037	[4.744, 5.463]	−0.071	0.212	[−0.099, −0.040]
Kul	0.1	5.057	0.023	[4.831, 5.299]	−0.067	0.152	[−0.085, −0.046]
	0.2	5.100	0.034	[4.743, 5.457]	−0.070	0.206	[−0.099, −0.039]
	0.25	5.098	0.038	[4.718, 5.468]	−0.070	0.234	[−0.100, −0.037]

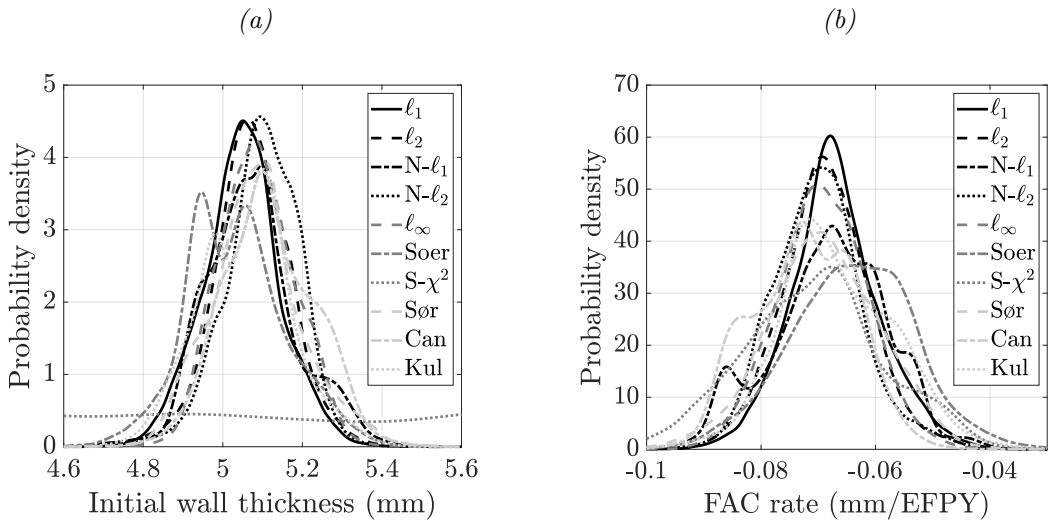


Figure 5.19: Marginal posterior distributions of the (a) initial wall thickness and (b) FAC rate obtained using $p_0 = 0.1$ under different distance settings.

rate. An important quantity in industrial maintenance planning is the lower bound of the system lifetime, which is found to be approximately 35 EFPY. This indicates that around 35 EFPY, the feeder pipes need to be replaced to continue safe operation of the nuclear power generation unit. The distributions of the system lifetimes obtained using the selected distance functions are shown in Figure 5.20a. The figure also confirms that the distance functions mentioned above provide best results from ABC-SS. It is often convenient and of more interest to practitioners to study the survival function of a system of components. Thus, the corresponding survival functions of the feeder pipes are shown in Figure 5.20b.

Component-Specific Characteristics

The advantage of using a mixed-effects regression model over a simple regression model is that one can easily estimate the component-specific degradation characteristics using the entire data set. In simple linear regression, component-specific degradation characteristics are calculated independently based only on the component-specific data, the

Table 5.8: Summary statistics of the lifetime distribution of feeder pipes.

Distance fn.	p_0	Mean (EFPY)	COV	95% CI (EFPY)
ℓ_1	0.1	**43.00**	**0.108**	**[35.70, 54.64]**
	0.2	41.58	0.187	[31.91, 59.03]
	0.25	42.56	0.207	[31.25, 62.43]
ℓ_2	0.1	**42.37**	**0.104**	**[35.49, 52.56]**
	0.2	43.03	0.153	[33.38, 58.19]
	0.25	42.55	0.173	[31.61, 60.29]
N-ℓ_1	0.1	42.91	0.122	[34.40, 54.10]
	0.2	44.60	0.240	[31.56, 71.26]
	0.25	44.82	0.294	[30.37, 80.19]
N-ℓ_2	0.1	**41.86**	**0.104**	**[35.19, 52.51]**
	0.2	41.89	0.173	[31.21, 58.90]
	0.25	42.56	0.175	[32.33, 60.83]
ℓ_∞	0.1	**42.19**	**0.102**	**[35.21, 52.09]**
	0.2	42.62	0.137	[33.59, 56.15]
	0.25	43.02	0.187	[32.82, 60.96]
Soer	0.1	45.27	0.158	[35.01, 62.25]
	0.2	43.83	0.226	[31.19, 68.66]
	0.25	44.78	0.278	[30.99, 77.50]
S-χ^2	0.1	41.45	0.193	[25.66, 56.76]
	0.2	45.80	0.172	[30.23, 61.56]
	0.25	45.54	0.214	[31.62, 63.48]
Sør	0.1	42.55	0.124	[34.32, 54.66]
	0.2	44.03	0.211	[32.49, 66.06]
	0.25	43.68	0.232	[31.15, 67.43]
Can	0.1	42.55	0.124	[34.32, 54.66]
	0.2	41.58	0.126	[33.46, 53.41]
	0.25	42.70	0.213	[31.50, 65.24]
Kul	0.1	43.77	0.142	[35.04, 58.62]
	0.2	42.89	0.223	[32.10, 64.66]
	0.25	43.48	0.245	[31.49, 68.01]

(a) *(b)*

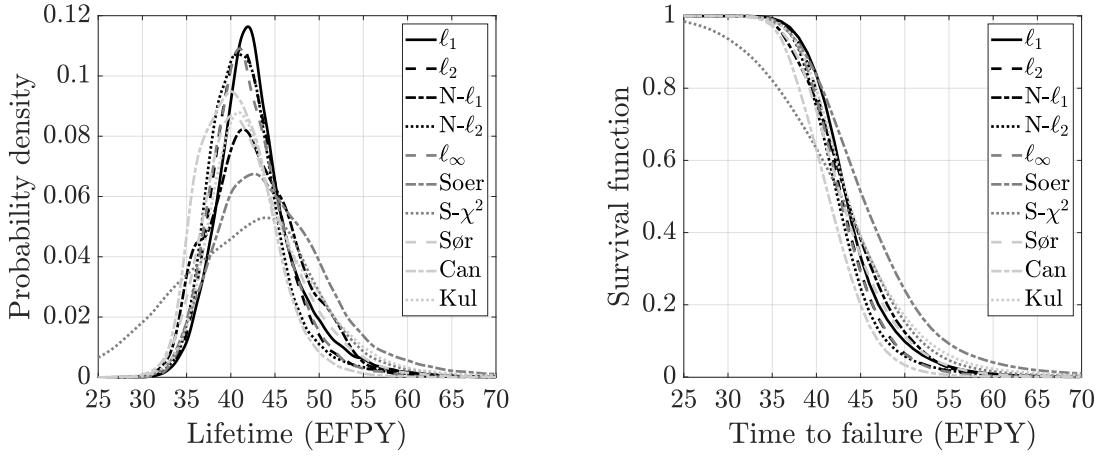


Figure 5.20: (a) Lifetime distribution and (b) survival function of the system of feeder pipes obtained using $p_0 = 0.1$ under different distance settings.

amount of which is comparatively quite small (1 to 5 data points in case of feeder pipes). As a result, the component-specific degradation parameters suffer from high uncertainties which creates difficulty in making maintenance decisions [94]. Figure 5.21 illustrates the variation in feeder-specific corrosion model parameters. The posterior

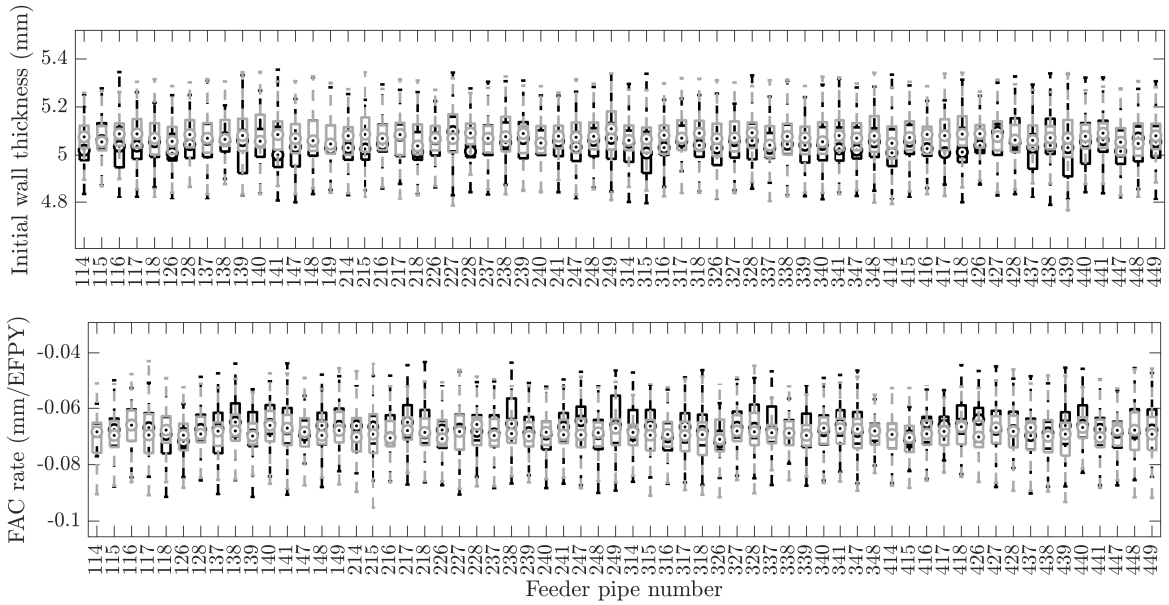


Figure 5.21: Box plots of the posterior distributions of component-specific initial wall thicknesses and FAC rates obtained using $p_0 = 0.1$. Box plots of samples using Manhattan (ℓ_1) distance are in black and Euclidean (ℓ_2) distance in gray.

distributions of feeder-specific initial wall thicknesses and FAC rates obtained from ABC-SS using the Manhattan (ℓ_1) and Euclidean (ℓ_2) distance functions are represented using box plots. Small circles with a dot inside them indicate the median values, and the top and bottom ends of the boxes indicate the 25th and 75th percentiles. The whiskers extend to the maximum and minimum data points considered after discarding the outliers (if any). The pipe numbers of 62 Type M feeders are indicated at the horizontal axis. The medians of the individual initial wall thicknesses fluctuate around 5.1 mm pipe thickness, whereas the medians of the individual corrosion rates are close to 0.07 mm/EFPY value. To further extend the feeder-specific analysis, the

feeder-specific corrosion parameters can be used to predict the survival probabilities of the individual feeders. For illustration, the survival functions of the individual feeders obtained using $p_0 = 0.1$ and Manhattan (ℓ_1) and Euclidean (ℓ_2) distance functions are presented in Figure 5.22. For comparison, the survival function of the system of feeder

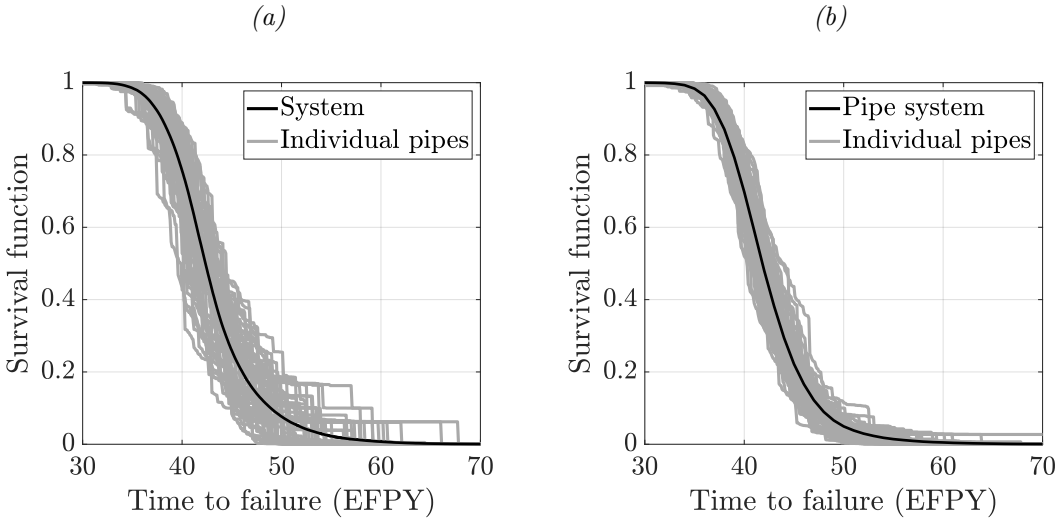


Figure 5.22: Survival functions of the feeder pipes obtained using (a) Manhattan (ℓ_1) and (b) Euclidean (ℓ_2) distance functions.

pipes are shown on top of the feeder-specific survival functions. It can be noticed that the ℓ_1 distance produces higher variability of individual component lifetimes compared to the results produced by the ℓ_2 distance.

Comparison with Simple Linear Regression

To illustrate the impact of using a simple linear regression model (i.e., using only the fixed effects parameters in the LMER model and ignoring the random effects parameters), Figure 5.23a and Figure 5.23b present the lifetime distributions and survival functions of the system of feeder pipes, respectively, obtained using both the models. The posterior samples are generated using ℓ_1 and ℓ_2 distance functions and by fixing $p_0 = 0.1$ in the ABC-SS algorithm. The figures show that the simple linear regression

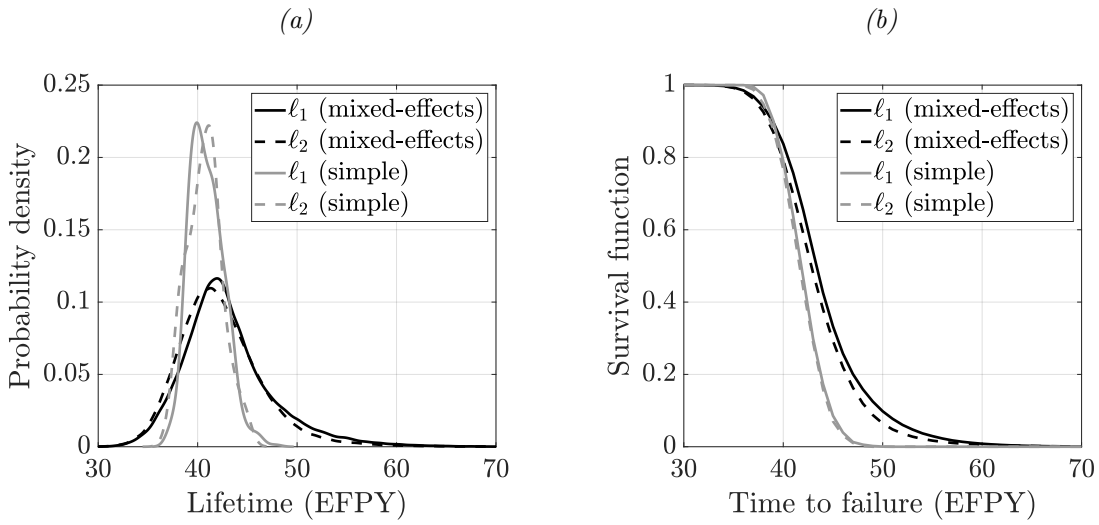


Figure 5.23: (a) Lifetime distributions and (b) survival functions of the system of feeder pipes obtained using LMER and simple linear regression models. The results are generated by setting $p_0 = 0.1$ and using ℓ_1 and ℓ_2 distance functions.

model captured less uncertainty of the feeder lifetime compared to the LMER model. From the simple linear regression model, the mean lifetime is found to be around 41 EFPY and COV around 4% – a reduction of approximately 60% from the COV of the LMER lifetime. Moreover, the simple linear regression model gives a lower bound of approximately 38 EFPY and an upper bound of 44 EFPY for the 95% CI of system lifetime. By comparison, the LMER model produces a lower bound of 35 EFPY and an upper bound of 53 EFPY. The reduction in credible bounds of the feeder lifetime clearly shows that the simple linear regression model is ineffective in accurately capturing the uncertainty of the system lifetime.

5.5 Concluding Remarks

Degradation data from nuclear power plants are generally limited and masked with measurement errors. As a result, stochastic modeling of degradation data becomes

computationally challenging. The reason is that large sampling and inspection uncertainties in degradation measurements require uncertainty quantification of the degradation model parameters. The most popular approach in this case is the Bayesian approach. Parameter inference of the model parameters in a Bayesian setting requires evaluation of the model likelihood several times. However, in real-life situations, the sample likelihoods of the model parameters derived from noisy data are complicated high-dimensional integrals – making the traditional likelihood-based Bayesian inference schemes computationally prohibitive.

Among the three examples presented in this chapter, the first two considered the data to be contaminated by only the sizing error, whereas the third example considered both the sizing and coverage error issues in degradation modeling. All three examples using real data sets proves that the proposed ABC approach can be implemented with confidence to completely bypass the likelihood evaluation step in the Bayesian inference procedure. The likelihood-free ABC methods offer high efficiency with minimal analytical and computational complexities. Thus, ABC promises to be a better and practical approach for solving real-life modeling problems that particularly involve various inspection uncertainties.

The first two examples highlighted the potential of ABC-MCMC to handle multiple parameter estimation problems in corrosion modeling. This approach generated results which match closely to that from the L-MCMC method. In the first example, the estimates of the FAC rate parameters obtained from the Bayesian approach are compared with the industry-standard linear regression approach. The results show that the linear regression approach may underestimate the lifetime of the feeder pipes, thereby reinforcing the need to adopt a Bayesian framework for the estimation of corrosion model parameters. The third example used the LMER model to characterize the FAC data, where the ABC-SS algorithm is used to estimate the model parameters. The study

shows that the proposed ABC-SS method efficiently handles the model calibration and prediction provided the algorithmic hyperparameters are selected properly.

Chapter 6

Parameter Estimation of a Localized Corrosion Model

6.1 Introduction

Nuclear power plants consist of several large and small structural components that work as a system to generate electricity. The reactor core contains numerous fuel channels that contain the nuclear fuel, whereas several hundred feeder pipes and the steam generators comprise of the primary heat transport system (see Figure 1.1). The heat generated in the reactor core is transported to the steam generators with the help of the heavy water coolant running through the feeder pipes that are connected to the fuel channels. The steam generators contain numerous thin-walled tubes that help in producing steam which is used by the steam turbines to generate electricity [77]. The steam generator tubes experience a high degree of pitting corrosion due to extreme conditions of the surrounding environment [148]. As a result, the outside surfaces of these tubes are susceptible to pipe leakage if the pit depths are left uninspected for a long period of time.

Pitting corrosion is a type of localized corrosion that starts by cavity formation on the metal surfaces [123]. These cavities can penetrate surfaces leading to failures. The process of pit generation and their growth is considered to be stochastic in nature [148]. In fact, due to this uncertainty, the new pit locations and growth rates of the existing pits are difficult to predict. To understand the process of pitting corrosion, many mechanistic (e.g., [88–90]) and stochastic models (e.g., [84, 117, 118, 148]) have been developed.

This chapter presents an example of modeling the pitting corrosion data obtained from in-service inspections of the steam generator tubes of a nuclear power plant. The main objective is to estimate the distribution of the maximum pit depth – a quantity of interest for life-cycle management of steam generators. The stochastic model for the pitting corrosion process proposed by Yuan et al. [148] is used in this example. The model deals with pit generation and growth in a systematic way, and it is fairly realistic since it considers both the measurement and detection errors in the pitting corrosion data.

Yuan et al. [148] showed that the likelihood function under the measurement and POD errors involves high-dimensional integrals and infinite summations (see Equation 2.31). Moreover, because of the POD function, the parameters of the pit generation process and the pit growth process become intertwined to a point that it is not possible to separately estimate them even though both processes are assumed independent. As a result, the standard MCMC-based Bayesian inference schemes fail to provide an efficient and accurate framework for parameter estimation. To circumvent the numerical difficulties with likelihood evaluation, Yuan et al. [148] considered using an approach based on the data augmentation technique – an iterative method that simulates missing data from updated model parameters [51]. Lu [77], on the other hand, tried to solve a similar problem using an approximate version of the likelihood function

which helps in reducing numerical difficulties. However, this chapter shows that the proposed ABC-based approach not only eliminates the need to evaluate the likelihood function but also provides an intuitive and efficient way for Bayesian inference. The subsequent section presents an application of the newly developed ABC-SS(MHMC) algorithm (see Subsection 4.9.3) for modeling pitting flaws in the steam generator tubes.

6.2 Example: Parameter Estimation and Prediction of Maximum Pit Depth

6.2.1 Degradation Data and Model

Data Set

The data set is prepared using information from a total of six inspection campaigns. It contains information only about the newly detected pitting flaws since majority of the pits do not show significant growths after detection [148]. The depths of the pitting corrosion flaws are measured using an eddy current probe. Table 6.1 summarizes the inspection data which show the number of newly detected pits along with the means and the standard deviations of measured pit depths from each inspection campaign. The pit depths are expressed in terms of the percentages of through-wall depth (TWD) of steam generator tubes. The maximum of the mean pit depths can be seen to be around 30% TWD, whereas the maximum of the standard deviation of pit depths is around 20%. The number of newly detected pits show an increasing trend over the years until the fifth inspection. Thereafter, at the sixth inspection campaign, it suddenly drops to a significantly small number of 18 from 238, which is the number of newly detected pits at the fifth inspection campaign. The reason is a major mainte-

nance campaign that is carried out just after the fifth inspection campaign. A major maintenance campaign generally includes water lancing (WL) and chemical cleaning (CC) of the steam generator tubes [148]. This example utilizes data from the first five inspection campaigns for calibrating the pitting corrosion model. The data from the sixth inspection campaign are utilized for model validation.

Table 6.1: Summary of pitting corrosion data from all six inspection campaigns.

Inspection campaign no.	Time (years)	No. of newly detected pits	Mean pit depth (% TWD)	Standard deviation of pit depths (% TWD)
1	1.5	87	20.8	10.7
2	2.5	26	19.7	5.1
3	6.4	49	30.3	1.9
4	7.5	123	18.9	2.7
5	8.3	238	13.2	2.9
6	12.3	18	23.8	20.3

Model

For convenience, the NHPP-Weibull model, presented in Section 2.2.3, is briefly described here. Assuming the inspections are carried out at times $t_1, t_2, \cdots, t_k$, the data $\mathbf{D}_i = \{n_{di}, \mathbf{h}_i^{(m)}\}$ from ith inspection contain the number of newly detected pits n_{di} and their measured depths $\mathbf{h}_i^{(m)} = \{h_{i1}^{(m)}, h_{i2}^{(m)}, \cdots, h_{i,n_{di}}^{(m)}\}^\top$. The actual number of pits can be written as $n_i = n_{di} + n_{ui}$, where n_{ui} is the number of undetected pits. The measured pit depths can be written as $\mathbf{h}_i^{(m)} = \mathbf{h}_i + \mathbf{z}_i$, where $\mathbf{h}_i = \{h_{i1}, h_{i2}, \cdots, h_{i,n_{di}}\}^\top$ are the actual pit depths and $\mathbf{z}_i = \{z_{i1}, z_{i2}, \cdots, z_{i,n_{di}}\}^\top$ are the measurement errors. The actual number of pits n_i is modeled using NHPP, thus follows the Poisson distribution: $P[N_i = n] = \frac{[\Lambda(t_{i-1},t_i)]^n}{n!} e^{-\Lambda(t_{i-1},t_i)}$, $n = 0, 1, 2, \cdots$, where $\Lambda(t_{i-1}, t_i) = \int_{t_{i-1}}^{t_i} \nu(t)dt = \lambda(t_i^\delta - t_{i-1}^\delta)$, $\nu(t) = \lambda\delta t^{\delta-1}$ is a power law intensity function, and $\lambda > 0$ and $\delta > 0$ are the parameters of the NHPP model. To characterize the detection error, the POD function $p(h) = 1 - \frac{1+e^{-qw}}{1+e^{q(h-w-t_h)}}$ is considered, which is valid if the pit depth $h > t_h$; otherwise, it is zero. In this example, the fol-

lowing parameter values are chosen: $q = 20$, $w = 0.1$, and $t_h = 0$, which are taken from reference [148]. The actual pit depths are assumed to be Weibull distributed as $f_H(h) = (\beta/\gamma)(h/\gamma)^{\beta-1}e^{-(h/\gamma)^{\beta}}$, $h > 0$, where $\gamma > 0$ and $\beta > 0$ are scale and shape parameters, respectively. The measurement errors are assumed to follow the normal distribution with a zero mean and a fixed standard deviation of 0.05 mm. For the derivation of the distribution of maximum pit depth, the reader is referred to Appendix E.

6.2.2 Implementation Details

The proposed ABC-SS(MHMC) algorithm (Subsection 4.9.3) is implemented in the MATLAB environment (version 9.9, 64-bit) with Intel® Core™ i5-1035G7 CPU @ 1.20 GHz processor, and 8.00 GB memory (RAM). To demonstrate the application of the model selection procedure (for details, see Section 4.10), the pit generation model is extended to four different models, $m = 1, 2, \cdots, 4$, by specifying $\delta = \{0.5, 1.0, 1.5, 2.0\}$. Let us limit ourselves to these four models although there could be many ways to extend the basic model. The three other parameters $\{\lambda, \gamma, \beta\}$ are estimated for each individual model.

Prior Distribution

Since all of the parameters of the pit generation and growth processes have positive supports, a computationally convenient choice is to consider the logarithmic versions of the parameters for parameter estimation, i.e., $\boldsymbol{\Theta} = \{\ln\lambda, \ln\gamma, \ln\beta\}$. For selecting the joint prior distribution, note that the gradient of a multi-variate Gaussian log-likelihood can be analytically derived, hence can be used in our advantage. Thus, a multi-variate Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is selected as the joint prior distribution. To select a non-informative diffused prior, the mean vector is assumed to be zero, whereas the

covariance matrix $\mathbf{\Sigma}$ is assumed to have diagonal terms equal to 10 (variances) and non-diagonal terms equal to zero.

Distance Function

In this example, the following distance function is chosen:

$$\rho(\mathbf{D}^*, \mathbf{D}) = \sum_i \frac{w_1|n^*_{di} - n_{di}|}{|n^*_{di}| + |n_{di}|} + \frac{w_2|m^*_i - m_i|}{|m^*_i| + |m_i|} + \frac{w_3|s^*_i - s_i|}{|s^*_i| + |s_i|} + \frac{w_4|sk^*_i - sk_i|}{|sk^*_i| + |sk_i|} \tag{6.1}$$

where m^*_i and m_i are the sample means of the simulated and actual measured pit depths at the ith inspection campaign, respectively; s^*_i and s_i are the sample standard deviations; sk^*_i and sk_i are the sample skewnesses; and $\{w_1, w_2, w_3, w_4\}$ are the weights.

The selection of this distance function can be justified as follows. The observed data contain two main information: the number of newly detected pits and their respective measured depths. Thus, the proposed ABC method needs a distance function that involves information from both types of data. Although the distance function in Equation 6.1 is selected in an ad hoc manner (since there is no practical guideline available in the literature [142]), this example proves that the proposed distance function is quite effective in capturing the target quantity of interest. While the discrepancy between number of pits are directly used in the selected distance function, the discrepancy between the measured and simulated pit depths is calculated using three summary statistics: mean, standard deviation, and skewness. To give equal weights to the number of pits and pit depth data, the following values are chosen for the weights: $w_1 = 0.5$ and $w_2 = w_3 = w_4 = 0.5/3$.

Selection of the Algorithmic Hyperparameters

To implement the algorithm, n_0 is fixed at 100,000 samples and p_0 is fixed at 0.1. The number of simulation levels is determined using an approach proposed by Hazra et al. [59], which states that the algorithm should be stopped after a certain number of simulation levels when less than a certain p% change is observed between the tolerance thresholds of the current simulation level and the previous simulation level. Here, the percent value is selected to be 3%. The number of leapfrog steps is selected to be $L = 10$, and the time step parameter is selected to be $h = 0.05$. The algorithmic hyperparameters are selected based on a few trial runs of the algorithm to achieve higher accuracy with less sample repetitions.

Data Simulation

The data simulation is performed by mimicking the process of pit generation and growth according to the underlying stochastic model. Using the updated model parameters, a simulated data set $\mathbf{D}_i^* = \{n_{di}^*, \mathbf{h}_i^{*(m)}\}$, $i = 1, 2, \cdots, k$ (the total number of inspections $k = 5$) is generated for each inspection campaign. The number of new pits n_i^* are simulated from the NHPP model defined in Equation 2.23, whereas the pit depths $\mathbf{h}_i^*$ are simulated from the Weibull distribution $f_H(h)$ defined in Equation 2.25. The actual pit depths $\mathbf{h}_i^*$ are made noisy by adding the noise terms $\mathbf{z}_i^*$ simulated from the Gaussian distribution $\mathcal{N}(0, 0.05^2)$. Thus, the measured pit depths are obtained as $\mathbf{h}_i^{*(m)} = \mathbf{h}_i^* + \mathbf{z}_i^*$. Next, a pit detection process is created so that different pits with their respective depths are either detected or not detected according to the POD function defined in Equation 2.26. The number of newly detected pits n_{di}^* and their measured depths $\{h_{i1}^{*(m)}, h_{i2}^{*(m)}, \cdots, h_{i,n_{di}^*}^{*(m)}\}$ represent the simulated data set $\mathbf{D}_i^*$ for ith inspection campaign. The same data simulation process is repeated for all five inspection campaigns.

6.2.3 Results and Discussion

Parameter Estimates

The proposed model selection algorithm (Section 4.10) is run for seven simulation levels and then it stopped once the stopping criterion is fulfilled. The algorithm took around 15 minutes to run. Figure 6.1 shows the histograms of models at different simulation levels starting from level 1 and ending at level 7. The simulation level 7 is the final

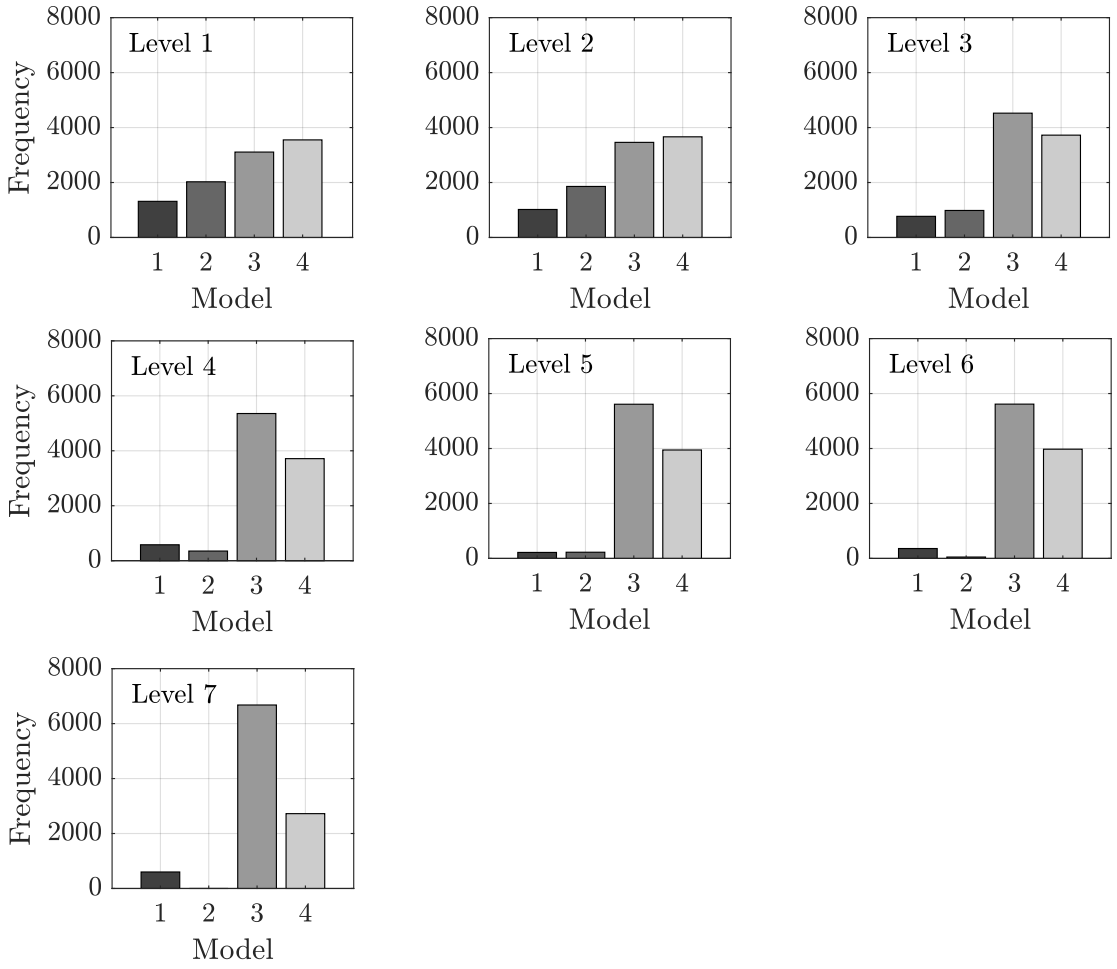


Figure 6.1: Histograms of models at different simulation levels.

level which represents the posterior estimates of the models. At each simulation level, 10000 sets of parameter samples are selected as seed samples for the next simulation

level. In Figure 6.1, slight variations of the frequencies of four different models can be observed at the first two levels; however, from level 3, the frequencies of the models start to change rapidly. It can be observed that models 1 and 2 almost vanished after level 4, whereas the frequency of model 3 gradually increases to a maximum at the final simulation level. At the final simulation level, out of 10000 sets of parameter samples, models $1 \sim 4$ were selected 600, 0, 6677, and 2723 times, respectively. Thus, the model selection scheme selects model 3 as the best model to represent the observed data among all four models. The Bayes factors $\mathcal{B}_{31} = 11.13$, and $\mathcal{B}_{34} = 2.45$ imply that model 3 shows positive evidence against model 1 and weak evidence against model 4. Although the scheme finds model 3 as the most suitable model for the data, it can be observed that model 3 is only marginally better than model 4.

The marginal posterior distributions of the model 3 parameters are shown in Figure 6.2. It shows that the model 3 parameters have fair amounts of variability that

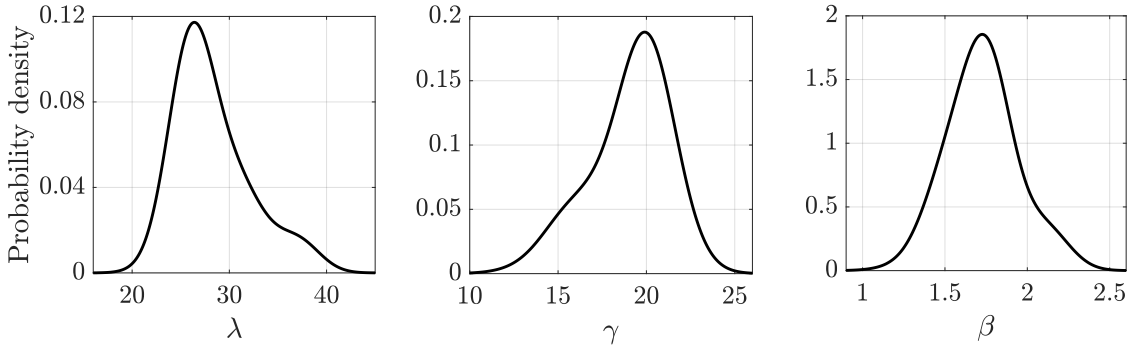


Figure 6.2: Marginal posterior distributions of the model parameters.

may lead to higher uncertainties in the model predictions. The means, COVs, and 95% credible intervals of the parameters are calculated and listed in Table 6.2. It can be observed that the parameters have similar COV values of around 15%. This implies that the uncertainties in the model predictions are contributed approximately equally by all three parameters of the model.

Table 6.2: Summary statistics of the posterior parameter samples.

Parameter	Mean	COV	[2.5th, 97.5th] percentiles
λ	28.27	0.15	[21.85, 38.25]
γ	19.03	0.13	[13.66, 23.05]
β	1.81	0.14	[1.29, 2.28]

Distribution of Pit Depth

A major maintenance of the steam generator tubes through the the WL/CC process is carried out at the end of the fifth inspection campaign. Thus, the sixth inspection campaign can be considered to be the first inspection of the steam generator tubes after a major maintenance campaign. Table 6.1 shows that the time interval between the fifth and sixth inspection campaigns is four years. In Figure 6.3, the distributions of the measured and predicted pit depths are compared. The histogram represents the measured pit depths. The solid gray lines represent the predicted pit depth distribu-

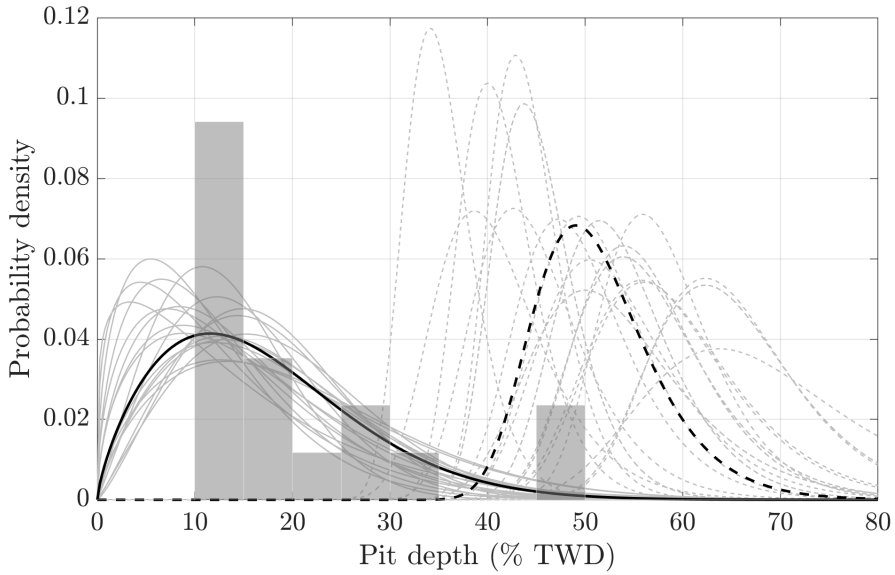


Figure 6.3: Comparison between measured and predicted pit depth distributions (obtained at an interval of four years) at the sixth inspection campaign. The histogram represents the measured/observed pit depths. The solid and broken gray lines represent the predicted distributions of pit depths and maximum depths for different sets of parameter samples. The solid and broken black lines represent the corresponding mean predicted distributions.

tions obtained using all sets of the posterior parameter samples from model 3; whereas the solid black line represents the predicted mean distribution. The predicted mean pit depth distribution can be seen to fit well with the observed distribution, which also includes the smaller pits that were otherwise missed due to the POD error. The same can be noticed in Figure 6.4, where the predicted mean pit depth distribution is compared with various other probability distributions, such as Weibull, gamma, and lognormal, fitted to the measured pit depths using the MLE method. The predicted distribution takes care of the POD and measurement errors compared to the fitted distributions that are based on the noisy data.

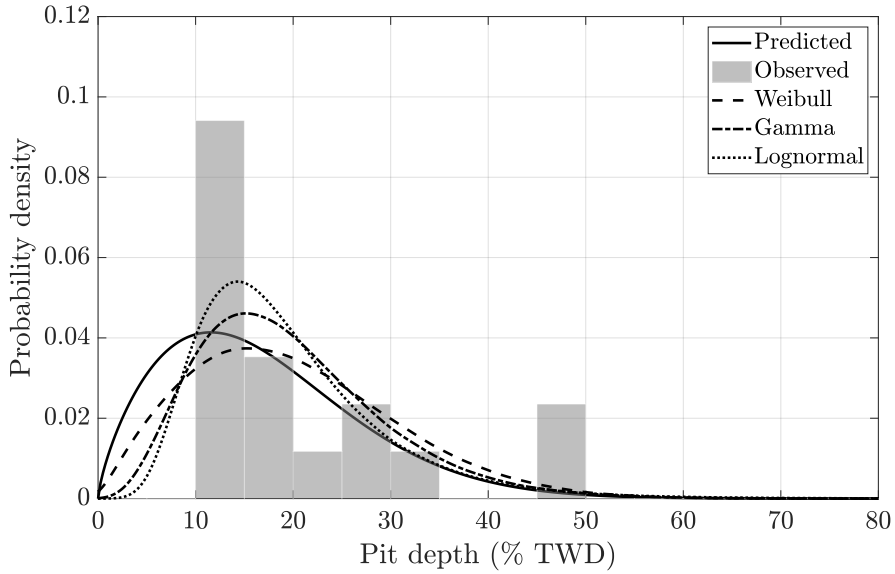


Figure 6.4: Comparison between the predicted pit depth distribution and various other probability distributions fitted to the measured/observed pit depth data.

Distribution of Maximum Pit Depth

It is often of great interest to a practitioner to determine the distribution of the maximum pit depth. The probability distribution of the maximum pit depth is plotted in the same Figure 6.3. The broken gray lines represent the predicted maximum pit depth

distributions generated using all instances of the posterior parameter samples, and the solid black line represents the predicted mean maximum pit depth distribution. From the figure, it can be visually verified that the observed maximum pit depth is covered by the predicted mean distribution. This proves that the proposed methodology can be used to accurately predict the distributions of the pit depth and its maximum value in the steam generator tubes.

The variability of the 95th percentile of maximum pit depth with the operating time of the steam generators is another important input to the risk-based life cycle management of nuclear power plant components. Thus, the posterior parameter samples are utilized in predicting the 95th percentile of maximum pit depth in Figure 6.5. The solid line represents the mean path and the shaded region represents the 95% credible interval. The figure shows that the 95th percentile of maximum pit depth increases rapidly in the first few years of operation and thereafter, it grows slowly along time.

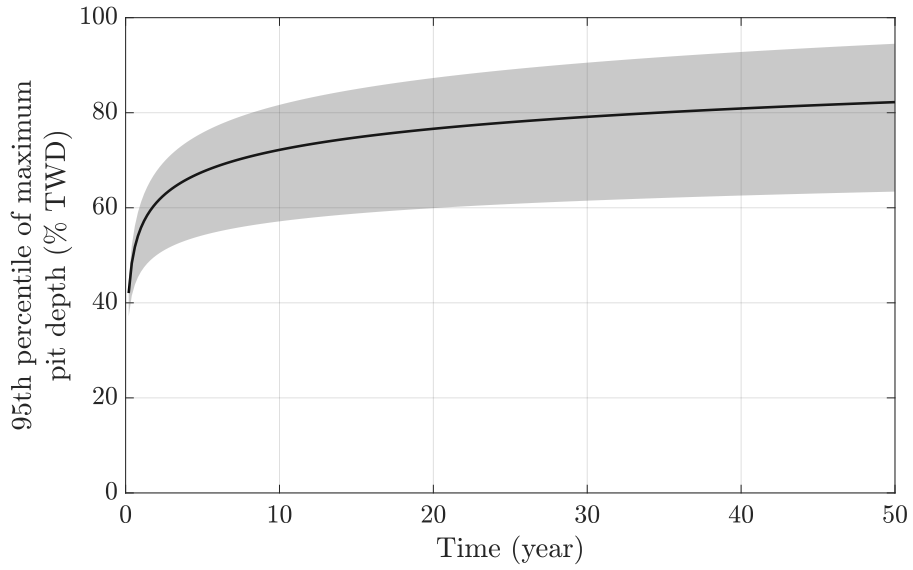


Figure 6.5: Prediction of the 95th percentile of maximum pit depth. The solid line represents the mean and the shaded region represents the 95% credible interval.

Comparison with the results of a previous study

As mentioned earlier, Yuan et al. [148] proposed an approach based on MCMC and the data augmentation technique, and here we compare Yuan's results with the results obtained using the likelihood-free approach. The parameter estimates from the study by Yuan et al. [148] are presented in Table 6.3. It can be observed from Table 6.2

Table 6.3: Posterior parameter estimates reported in [148].

Parameter	Mean	COV
λ	23.8	0.20
γ	18.1	0.10
β	1.47	0.05
δ	1.79	0.09

that the means and COVs of the posterior parameter samples from both methods are almost similar, and the slight variations are the results of selecting different priors for the Bayesian inference. While the prior selection in the study by Yuan et al. [148] was motivated by prior conjugacy, in this study, the prior selection is mainly motivated by analytical and numerical convenience. However, if one compares the model predictions, it can be observed that both methods produce similar results (see Figure 7 in [148]).

6.3 Concluding Remarks

Modeling pitting corrosion data and predicting the maximum pit depth distribution is a challenging task since the data are generally limited and mostly affected by sizing and detection errors. Although the parameter uncertainties can be suitably quantified in a Bayesian framework, the conventional likelihood-based schemes simply do not qualify as practical methods due to difficulty in evaluating the model likelihood. On the other hand, the proposed ABC approach proves to handle the modeling and prediction process quite efficiently.

The proposed ABC-SS(MHMC) scheme is applied to model selection and parameter estimation of the pitting corrosion model defined in Section 2.2.3. The scheme successfully identified the most suitable model that best describes the observed data, and in addition to that, it estimated the corresponding model parameters. The posterior parameter samples are then used to predict the distribution of maximum pit depth and the 95th percentile of maximum pit depth along time, which are important inputs to the life cycle management of steam generators.

Chapter 7

Conclusions and Recommendations

7.1 Significance of the Study

The reliability of an infrastructure system is adversely affected by various degradation processes over time. As a result, the reliability of these kinds of systems reduces over time, and if not maintained properly, they can fail even before reaching their end of life. This study presents a likelihood-free approach for Bayesian degradation modeling using degradation measurements that mainly suffer from various uncertainties related to measurement noise and detection errors. The significance of the proposed research mainly lies in the fact that it tried to explore the effects of parameter uncertainty in model predictions in a simple and efficient way. To this end, this study has provided an easy-to-implement method for Bayesian model selection and parameter inference of stochastic degradation models. A benefit of the proposed method is that it expands the domain of model choice, thereby allowing practitioners to develop more realistic models without being constrained by the analytical complexities.

7.2 Summary of Results

Four popular stochastic degradation models for flaw growth and flaw generation are introduced in Chapter 2, these are the random rate model, the gamma process model, the LMER model, and the Poisson process model. The basic properties of these models and the corresponding model likelihoods for noisy data are presented in this chapter. The common sizing error is considered in the formulation of the random rate and gamma process models. In the formulation of the LMER model, besides sizing error, the coverage error is considered, which mainly impacts inspection data of nuclear power plants. Lastly, the Poisson process flaw generation model is formulated for data contaminated by the sizing and POD errors.

For Bayesian inference, the MCMC methods are popular since they can efficiently generate samples from a posterior distribution without having to calculate the entire posterior density function. These methods use only a ratio of the posterior density functions for sampling. The derivation of MCMC methods using the Markov chain theory in conjunction with the Monte Carlo Methods is discussed in Chapter 3. The basic implementation steps of various popular MCMC algorithms are presented along with a few toy examples to demonstrate the working principles of these schemes.

The study finds that MCMC methods become computationally prohibitive when degradation measurements are noisy. This happens because noisy data often turns model likelihoods into various complicated functions, such as high-dimensional integrals, high-dimensional infinite summations, or a combination of both. Numerical evaluation of these functions is challenging since most methods suffer from convergence issues. As a remedy to this problem, the likelihood-free ABC schemes for Bayesian inference are introduced in Chapter 4. The fundamental idea behind the likelihood-free inference and various implementation issues of the advanced ABC algorithms are dis-

cussed in this chapter. An advanced ABC algorithm is proposed in this chapter, which is derived using the HMC method as its sampling scheme. It is shown using a toy example that the proposed ABC-HMC provides better mixing and exploration of small modes than the standard ABC-MCMC scheme. A new modified ABC-HMC algorithm is also proposed to accommodate highly diffused priors, which uses the subset simulation as a sequential sampling scheme within it.

To demonstrate the capabilities of the proposed methodology, various examples of flaw growth and flaw generation modeling using real data sets are presented in Chapter 5 and Chapter 6. The examples show that the proposed ABC-based approach generates accurate posterior estimates, and it is computationally efficient compared to the traditional likelihood-based approach.

The main contributions of this thesis are summarized as follows:

1. For estimating the parameters of the popular random rate model from noisy data, a likelihood-free approach based on the ABC-MCMC scheme is presented. To initialize the ABC-MCMC algorithm, a new scheme Algorithm 13 is proposed. It is shown that the ABC-MCMC method combined with the proposed initialization scheme is computationally faster than the traditional likelihood-based MCMC approach. The estimates from the Bayesian approach are compared with the standard linear regression approach. The results show that the linear regression approach underestimates the lifetime of the feeder pipes.

2. Bayesian inference of the gamma process parameters from noisy degradation measurements is performed through a novel application of the ABC-MCMC method. A new tolerance selection criterion is proposed, which provides a better way to perform quality checks on the accepted samples generated by the ABC-MCMC scheme. This guarantees that the accepted samples belong to a distribution very

close to the true posterior distribution. Although ABC-MCMC offers significant computational savings, it is found that the samples from the Markov chain of the proposed method show very high repetition. To reduce its effects, it is recommended to select large thinning intervals for the ABC-MCMC samples.

3. An application of the mixed-effects regression model for characterizing the feeder pipe data is presented. This study considered both sizing and coverage error issues during the inspection and proposed a sequential ABC method, ABC-SS, for estimating the parameters of the regression model. The proposed method is found to be very useful to deal with flexible regression models, i.e., a regression model with different types of error distributions such as normal, non-normal, and mixture distributions. Various implementation issues are discussed in detail – particularly, the selection of the distance functions and other algorithmic hyperparameters in the context of degradation modeling. Furthermore, a new stopping criterion is proposed for the algorithm which is based on the tolerance thresholds obtained from each simulation level. The proposed methodology can be implemented to accurately infer estimates of the system and individual component-specific lifetimes and survival probabilities.

4. Although the ABC-MCMC sampler works faster than any likelihood-based approach, it comes with poor mixing properties that result in high sample repetition and subsequent removal of a large number of samples through thinning. To improve the mixing properties, a new ABC algorithm is derived based on the HMC sampler. The HMC sampling scheme is an MCMC method that follows the Hamiltonian dynamics to propose new samples from seed samples. Its non-random walk behavior help to explore the target probability space more effectively and efficiently than the standard random-walk MCMC method. The

convergence of the proposed ABC-HMC algorithm is proved by satisfying the detailed balance equation, and its efficacy is verified using a numerical example in Section 4.9.2. The example shows that the new ABC-HMC scheme can effectively capture small modes of a target posterior distribution with better mixing properties than the standard ABC-MCMC sampling scheme.

5. A new sequential ABC algorithm is proposed to deal with highly diffused priors in a Bayesian inference problem. The proposed ABC algorithm is based on the subset simulation method and a modified HMC algorithm. With faster convergence, the new ABC-SS(MHMC) sampler turned out to be a powerful method to sample from a complex multi-modal target density (see, for example, Section 4.9.3). The applicability of the proposed ABC-SS(MHMC) algorithm is further extended by transforming it into a likelihood-free Bayesian model selection tool in Algorithm 12. The application of the proposed algorithm is demonstrated using a practical data set from the steam generator tubes affected by pitting corrosion. The data are assumed to be contaminated by measurement and detection errors. The proposed scheme successfully identified the most suitable model that best describes the observed data, and in addition to that, it estimated the corresponding model parameters that are used to predict the maximum pit depth distribution with time – a quantity of interest for the life cycle management of the steam generator tubes.

7.3 Recommendations for Future Research

The degradation models considered in this study only tried to characterize the variation of a particular degradation process with respect to the operating time. However, the component degradation may vary with respect to other parameters of the surrounding

environment as well. Thus, other stochastic models that consider variables such as temperature, pressure, flow velocity, and pH value must be investigated in order to not only have a better understanding of degradation processes but also verify the efficacy of the likelihood-free schemes for model selection and parameter estimation. Moreover, this study focused only on flaw growth and flaw generation modeling problems. However, other types of degradation processes such as two-phase degradation modeling problems (e.g., [106]) must be investigated as well.

The motivation behind using the ABC algorithms for stochastic degradation modeling is that these algorithms are likelihood-free; thus, one can avoid computation of the complicated likelihood functions resulting from noisy degradation measurements. This also makes these algorithms quite practical to use. However, the main drawback of these algorithms is that there are no theoretical justification or practical guidelines available in the literature on how to select the distance function and other different algorithmic hyperparameters for modeling different types of degradation processes. Thus, in the context of stochastic degradation modeling, further theoretical and numerical investigations are needed to remedy these drawbacks found in the novel likelihood-free ABC schemes.

Bibliography

[1] M. Abdel-Hameed. A gamma wear process. *IEEE transactions on Reliability*, 24 (2):152–153, 1975.

[2] A. B. Abdessalem, N. Dervilis, D. Wagg, K. Worden, et al. ABC-NS: a new computational inference method applied to parameter estimation and model selection in structural dynamics. In *Congrès français de mécanique*. AFM, Association Française de Mécanique, 2017.

[3] A. B. Abdessalem, N. Dervilis, D. Wagg, and K. Worden. Model selection and parameter estimation in structural dynamics using approximate Bayesian computation. *Mechanical Systems and Signal Processing*, 99:306–325, 2018.

[4] A. B. Abdessalem, N. Dervilis, D. Wagg, and K. Worden. Model selection and parameter estimation of dynamical systems using a novel variant of approximate Bayesian computation. *Mechanical Systems and Signal Processing*, 122:364–386, 2019.

[5] J. Akeret, A. Refregier, A. Amara, S. Seehars, and C. Hasner. Approximate Bayesian computation for forward modeling in cosmology. *Journal of Cosmology and Astroparticle Physics*, 2015(08):043, 2015.

[6] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[7] S.-K. Au and J. L. Beck. Estimation of small failure probabilities in high dimen-

sions by subset simulation. *Probabilistic engineering mechanics*, 16(4):263–277, 2001.

[8] J. Bakker and J. van Noortwijk. Inspection validation model for life-cycle analysis. In *Bridge maintenance, safety, management and cost, proceedings of the second international conference on bridge maintenance, safety and management (IABMAS), Kyoto, Japan*, pages 18–22, 2004.

[9] M. Baragatti, A. Grimaud, and D. Pommeret. Likelihood-free parallel tempering. *Statistics and Computing*, 23(4):535–549, 2013.

[10] C. P. Barnes, D. Silk, and M. P. Stumpf. Bayesian design strategies for synthetic biology. *Interface focus*, 1(6):895–908, 2011.

[11] M. A. Beaumont. Approximate Bayesian computation in evolution and ecology. *Annual review of ecology, evolution, and systematics*, 41:379–406, 2010.

[12] M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

[13] M. A. Beaumont, J.-M. Cornuet, J.-M. Marin, and C. P. Robert. Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990, 2009.

[14] P. Berge, J. Ducreux, and P. Saint-Paul. Effects of chemistry on corrosion-erosion of steels in water and wet steam. In *Water Chemistry of Nuclear Reactor Systems 2*, pages 19–23. Thomas Telford Publishing, 1981.

[15] F. Berger and K.-F.-L. Hau. Mass transfer in turbulent pipe flow measured by the electrochemical method. *International Journal of Heat and Mass Transfer*, 20(11):1185–1194, 1977.

[16] J. M. Bernardo. Reference posterior distributions for bayesian inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 113–147, 1979.

[17] Z. Birnbaum and S. C. Saunders. A statistical model for life-length of materials. *Journal of the American Statistical Association*, 53(281):151–160, 1958.

[18] M. G. Blum, M. A. Nunes, D. Prangle, and S. A. Sisson. A comparative review of dimension reduction methods in approximate bayesian computation. *Statistical Science*, 28(2):189–208, 2013.

[19] N. Bousquet, M. Fouladirad, A. Grall, and C. Paroissin. Bayesian gamma processes for optimizing condition-based maintenance under uncertainty. *Applied Stochastic Models in Business and Industry*, 31(3):360–379, 2015.

[20] G. E. Box and G. C. Tiao. *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons, 2011.

[21] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[22] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov chain Monte Carlo*. CRC press, 2011.

[23] A. Buchholz and N. Chopin. Improving approximate Bayesian computation via quasi-Monte Carlo. *Journal of Computational and Graphical Statistics*, 28(1): 205–219, 2019.

[24] K. Burrill. Modeling flow-accelerated corrosion in CANDU. Technical report, Atomic Energy of Canada Limited (AECL), 1995.

[25] K. Burrill and E. Cheluget. Corrosion of CANDU outlet feeder pipes. Technical report, Atomic Energy of Canada Limited (AECL), 1999.

[26] F. Camacho and S. Pagan. Statistical methods for Darlington steam generator tube fitness-for-service assessments, Jul 2006.

[27] P. Chen, Z.-S. Ye, and Q. Zhai. Parametric analysis of time-censored aggregate lifetime data. *IISE Transactions*, 52(5):516–527, 2020.

[28] T. Cheng and M. Pandey. An accurate analysis of maintenance cost of structures experiencing stochastic degradation. *Structure and Infrastructure Engineering*, 8 (4):329–339, 2012.

[29] S. Chenouri. STAT440/840/CM761: Computational Inference Winter 2019. *Instructor*, 2019.

[30] M. Chiachio, J. L. Beck, J. Chiachio, and G. Rus. Approximate Bayesian computation by subset simulation. *SIAM Journal on Scientific Computing*, 36(3): A1339–A1358, 2014.

[31] S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algorithm. *The american statistician*, 49(4):327–335, 1995.

[32] J. D. Christopher, N. T. Wimer, C. Lapointe, T. R. Hayden, I. Grooms, G. B. Rieker, and P. E. Hamlington. Parameter estimation for complex thermal-fluid flows using approximate Bayesian computation. *Physical Review Fluids*, 3(10): 104602, 2018.

[33] H.-S. Chung. A review of CANDU feeder wall thinning. *Nuclear Engineering and Technology*, 42(5):568–575, 2010.

[34] E. Cinlar. *Introduction to stochastic processes*. Courier Corporation, 2013.

[35] E. Cinlar, Z. P. Bazant, and E. Osman. Stochastic process for extrapolating concrete creep. *ASCE J Eng Mech Div*, 103(6):1069–1088, 1977.

[36] E. Crema, K. Edinborough, T. Kerig, and S. Shennan. An approximate Bayesian computation approach for inferring patterns of cultural evolutionary change. *Journal of Archaeological Science*, 50:160–170, 2014.

[37] K. Csilléry, M. G. Blum, O. E. Gaggiotti, and O. François. Approximate Bayesian computation (ABC) in practice. *Trends in ecology & evolution*, 25(7):410–418, 2010.

[38] S. V. Datla, M. I. Jyrkama, and M. D. Pandey. Probabilistic modelling of steam generator tube pitting corrosion. *Nuclear Engineering and Design*, 238(7):1771–1778, 2008.

[39] P. Del Moral, A. Doucet, and A. Jasra. An adaptive sequential Monte Carlo

method for approximate Bayesian computation. *Statistics and Computing*, 22 (5):1009–1020, 2012.

[40] M. M. Deza and E. Deza. Encyclopedia of distances. In *Encyclopedia of distances*, pages 1–583. Springer, 2009.

[41] P. Diaconis. The Markov chain Monte Carlo revolution. *Bulletin of the American Mathematical Society*, 46(2):179–205, 2009.

[42] R. Dooley and V. Chexal. Flow-accelerated corrosion of pressure vessels in fossil plants. *International Journal of Pressure Vessels and Piping*, 77(2-3):85–90, 2000.

[43] C. C. Drovandi and A. N. Pettitt. Estimation of parameters for macroparasite population evolution using approximate Bayesian computation. *Biometrics*, 67 (1):225–233, 2011.

[44] J. Ducreux. The influence of flow velocity on the corrosion-erosion of carbon steel in pressurized water. In *Water chemistry of nuclear reactor systems 3 vol. 1. Proceedings of an international conference organised by the British Nuclear Energy Society and co-sponsored by the Institution of Chemical Engineers and the Royal Society of Chemistry, Bournemouth, 17-21 October, 1983*, 1983.

[45] L. Fenyvesi, H. Lu, and T. Jack. Prediction of corrosion defect growth on operating pipelines. In *2004 International Pipeline Conference*, pages 225–230. American Society of Mechanical Engineers, 2004.

[46] D. M. Frangopol, M.-J. Kallen, and J. M. v. Noortwijk. Probabilistic models for life-cycle performance of deteriorating structures: review and future directions. *Progress in Structural Engineering and Materials*, 6(4):197–212, 2004.

[47] W. J. Garland. The Essential CANDU – A Textbook on the CANDU Nuclear Power Plant Technology. *Chap 6*, 17:18, 2016.

[48] A. E. Gelfand and A. F. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–

409, 1990.

[49] A. Gelman. Prior distribution. *Encyclopedia of environmetrics*, 3(4):1634–1637, 2002.

[50] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.

[51] A. Gelman, H. S. Stern, J. B. Carlin, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

[52] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[53] C. J. Geyer. Practical Markov chain Monte Carlo. *Statistical science*, pages 473–483, 1992.

[54] C. J. Geyer. Markov chain Monte Carlo lecture notes. *Course notes, Spring Quarter*, 1998.

[55] W. R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC, 1995.

[56] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. Introducing Markov chain Monte Carlo. *Markov chain Monte Carlo in practice*, 1:19, 1996.

[57] M. S. Hamada, A. Wilson, C. S. Reese, and H. Martz. *Bayesian reliability*. Springer Science & Business Media, 2008.

[58] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. 1970.

[59] I. Hazra and M. D. Pandey. A likelihood-free approach towards bayesian modeling of degradation growths using mixed-effects regression. *Computers & Structures*, 244:106427, 2021.

[60] I. Hazra, M. D. Pandey, and M. I. Jyrkama. Estimation of flow-accelerated

corrosion rate in nuclear piping system. *Journal of Nuclear Engineering and Radiation Science*, 6(1):011106, 2020.

[61] I. Hazra, M. D. Pandey, and N. Manzana. Approximate Bayesian computation (ABC) method for estimating parameters of the gamma process using noisy data. *Reliability Engineering & System Safety*, 198:106780, 2020.

[62] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401, 1999.

[63] H. Hong. Application of the stochastic process to pitting corrosion. *Corrosion*, 55(1):10–16, 1999.

[64] P. Hougaard. Survival models for heterogeneous populations derived from stable distributions. *Biometrika*, 73(2):387–396, 1986.

[65] L. Huyse and A. van Roodselaar. Effects of inline inspection sizing uncertainties on the accuracy of the largest features and corrosion rate statistics. In *2010 8th International Pipeline Conference*, pages 403–413. American Society of Mechanical Engineers, 2010.

[66] E. Ishida, S. Vitenti, M. Penna-Lima, J. Cisewski, R. de Souza, A. Trindade, E. Cameron, V. Busti, C. collaboration, et al. Cosmoabc: likelihood-free inference via population Monte Carlo approximate Bayesian computation. *Astronomy and Computing*, 13:1–11, 2015.

[67] E. T. Jaynes. Prior probabilities. *IEEE Transactions on systems science and cybernetics*, 4(3):227–241, 1968.

[68] E. Jennings and M. Madigan. astroABC: an approximate Bayesian computation sequential Monte Carlo sampler for cosmological parameter estimation. *Astronomy and Computing*, 19:16–22, 2017.

[69] M. I. Jyrkama and M. D. Pandey. Methodology for predicting flow-accelerated corrosion wear using unreferenced multiple inspection data. *Nuclear Engineering*

and Design, 250:317–325, 2012.

[70] M. I. Jyrkama and M. D. Pandey. Quantifying probe coverage error in feeder thinning assessment. In *Proceedings of the 37th Annual CNS Conference, June 4-7, Niagara Falls, ON, Canada*, 2017.

[71] M.-J. Kallen and J. M. van Noortwijk. Optimal maintenance decisions under imperfect inspection. *Reliability engineering & system safety*, 90(2-3):177–185, 2005.

[72] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[73] R. E. Kass and L. Wasserman. Formal rules for selecting prior distributions: A review and annotated bibliography. *Journal of the American Statistical Association*, 1994.

[74] J. Lawless and M. Crowder. Covariates and random effects in a gamma process model with application to degradation and failure. *Lifetime Data Analysis*, 10 (3):213–227, 2004.

[75] R. A. Levine, Z. Yu, W. G. Hanley, and J. J. Nitao. Implementing componentwise Hastings algorithms. *Computational statistics & data analysis*, 48(2):363–389, 2005.

[76] J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[77] D. Lu. *Estimation of Stochastic Degradation Models Using Uncertain Inspection Data*. PhD thesis, University of Waterloo, Waterloo, Canada, 2012. URL http://hdl.handle.net/10012/6747.

[78] D. Lu, M. Pandey, and M. Jyrkama. Probabilistic estimation of flow-accelerated corrosion rate at the welded joints of the nuclear piping system. In *ASME 2012 Pressure Vessels and Piping Conference*, pages 311–317. American Society of

Mechanical Engineers, 2012.

[79] D. Lu, M. D. Pandey, and W.-C. Xie. An efficient method for the estimation of parameters of stochastic gamma process from noisy degradation measurements. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 227(4):425–433, 2013.

[80] S. Madanat and W. H. W. Ibrahim. Poisson regression models of infrastructure transition probabilities. *Journal of Transportation Engineering*, 121(3):267–272, 1995.

[81] S. Madanat, R. Mishalani, and W. H. W. Ibrahim. Estimation of infrastructure transition probabilities from condition rating data. *Journal of infrastructure systems*, 1(2):120–125, 1995.

[82] J.-M. Marin, P. Pudlo, C. P. Robert, and R. J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

[83] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26): 15324–15328, 2003.

[84] A. Martinsek. Reliable inference for the maximum pit depth within pitting colonies on long pipelines. *Corrosion*, 59(12):1058–1063, 2003.

[85] P. Mason. Approximate Bayesian computation of the occurrence and size of defects in advanced gas-cooled nuclear reactor boilers. *Reliability Engineering & System Safety*, 146:21–25, 2016.

[86] E. Meeds and M. Welling. GPS-ABC: Gaussian process surrogate approximate Bayesian computation. *arXiv preprint arXiv:1401.2838*, 2014.

[87] E. Meeds, R. Leenders, and M. Welling. Hamiltonian ABC. *arXiv preprint arXiv:1503.01916*, 2015.

[88] R. E. Melchers. Representation of uncertainty in maximum depth of marine

corrosion pits. *Structural safety*, 27(4):322–334, 2005.

[89] R. E. Melchers. Statistical characterization of pitting corrosion—part 1: Data analysis. *Corrosion*, 61(7):655–664, 2005.

[90] R. E. Melchers. Statistical characterization of pitting corrosion—part 2: Probabilistic modeling for maximum pit depth. *Corrosion*, 61(8):766–777, 2005.

[91] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[92] S. P. Meyn, R. L. Tweedie, et al. Computable bounds for geometric convergence rates of Markov chains. *The Annals of Applied Probability*, 4(4):981–1011, 1994.

[93] T. Micevski, G. Kuczera, and P. Coombes. Markov model for storm water pipe deterioration. *Journal of infrastructure systems*, 8(2):49–56, 2002.

[94] J. Mikko, P. Mahesh, and L. Ming. Estimating degradation growth rate and time of component replacement from limited inspection data using mixed-effects modelling. *SMiRT-25*, Charlotte, NC, USA, 2019.

[95] R. M. Neal et al. MCMC using Hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

[96] K. T. Nguyen, M. Fouladirad, and A. Grall. Model selection for degradation modeling and prognosis with health monitoring data. *Reliability Engineering & System Safety*, 169:105–116, 2018.

[97] L. Pacchiardi, P. Künzli, M. Schöngens, B. Chopard, and R. Dutta. Distance-learning for approximate Bayesian computation to model a volcanic eruption. *Sankhya B*, pages 1–30, 2020.

[98] M. Pandey and D. Lu. Estimation of parameters of degradation growth rate distribution from noisy measurement data. *Structural Safety*, 43:60–69, 2013.

[99] M. Pandey, X.-X. Yuan, and J. Van Noortwijk. The influence of temporal un-

certainty of deterioration on life-cycle management of structures. *Structure and Infrastructure Engineering*, 5(2):145–156, 2009.

[100] M. Pandey, D. Lu, and D. Komljenovic. The impact of probabilistic modeling in life-cycle management of nuclear piping systems. *Journal of Engineering for Gas Turbines and Power*, 133(1):012901, 2011.

[101] A. Papoulis and S. U. Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[102] P. C. Paris. A rational analytic theory of fatigue. *The trend in engineering*, 13: 9, 1961.

[103] M. Park, W. Jitkrittum, and D. Sejdinovic. K2-ABC: Approximate Bayesian computation with kernel embeddings. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 398–407. Proceedings of Machine Learning Research, 2016.

[104] R. D. Peng. *Advance Statistical Computing*. 2018.

[105] S. R. Prabhu, M. D. Pandey, N. Christodoulou, and B. W. Leitch. A surrogate model for the 3D prediction of in-service deformation in CANDU® fuel channels. *Nuclear Engineering and Design*, 369:110871, 2020.

[106] G. Prakash and S. Narasimhan. Bayesian two-phase gamma process model for damage detection and prognosis. *Journal of Engineering Mechanics*, 144(2): 04017158, 2018.

[107] D. Prangle. Lazy ABC. *Statistics and Computing*, 26(1-2):171–185, 2016.

[108] J. K. Pritchard, M. T. Seielstad, A. Perez-Lezaun, and M. W. Feldman. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798, 1999.

[109] H. Qin, W. Zhou, and S. Zhang. Bayesian inferences of generation and growth of corrosion defects on energy pipelines based on imperfect inspection data. *Re-*

liability Engineering & System Safety, 144:334–342, 2015.

[110] C. P. Robert. Approximate Bayesian Computation: A survey on recent results. In *Monte Carlo and Quasi-Monte Carlo Methods*, pages 185–205. Springer, 2016.

[111] G. O. Roberts and J. S. Rosenthal. Markov-chain Monte Carlo: Some practical implications of theoretical results. *Canadian Journal of Statistics*, 26(1):5–20, 1998.

[112] G. O. Roberts and J. S. Rosenthal. Markov chain Monte Carlo. 2003.

[113] G. O. Roberts, J. S. Rosenthal, et al. General state space Markov chains and MCMC algorithms. *Probability surveys*, 1:20–71, 2004.

[114] D. B. Rubin. Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, pages 1151–1172, 1984.

[115] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons, 2016.

[116] M. Sadegh and J. A. Vrugt. Approximate Bayesian computation using markov chain monte carlo simulation: DREAM (ABC). *Water Resources Research*, 50 (8):6767–6787, 2014.

[117] P. Scarf, R. Cottis, and P. Laycock. Extrapolation of extreme pit depths in space and time using the r deepest pit depths. *Journal of the Electrochemical Society*, 139(9):2621, 1992.

[118] T. Shibata. 1996 wr whitney award lecture: Statistical and stochastic approaches to localized corrosion. *Corrosion*, 52(11):813–830, 1996.

[119] S. Sisson, Y. Fan, and M. Tanaka. A note on backward kernel choice for sequential Monte Carlo without likelihoods. Technical report, Technical report, University of New South Wales, 2008.

[120] S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765,

2007.

[121] J. P. Slade and T. S. Gendron. Flow accelerated corrosion and cracking of carbon steel piping in primary water-operating experience at the Point Lepreau Generating Station. In *Proceedings from the 12th International Conference on Environmental Degradation of Materials in Nuclear Systems-Water Reactors, Salt Lake City, UT*, pages 773–782, 2005.

[122] A. F. Smith and G. O. Roberts. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(1):3–23, 1993.

[123] H.-H. Strehblow and P. Marcus. Mechanisms of pitting corrosion. *Corrosion mechanisms in theory and practice*, pages 201–238, 1995.

[124] M. Sunnåker, A. G. Busetto, E. Numminen, J. Corander, M. Foll, and C. Dessimoz. Approximate Bayesian computation. *PLoS computational biology*, 9(1): e1002803, 2013.

[125] M. A. Tanner and W. H. Wong. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540, 1987.

[126] S. Tavaré, D. J. Balding, R. C. Griffiths, and P. Donnelly. Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518, 1997.

[127] L. Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

[128] T. Toni and M. P. Stumpf. Simulation-based model selection for dynamical systems in systems and population biology. *Bioinformatics*, 26(1):104–110, 2009.

[129] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. Stumpf. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2008.

[130] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. Stumpf. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009.

[131] M.-N. Tran, D. J. Nott, and R. Kohn. Variational Bayes with intractable likelihood. *Journal of Computational and Graphical Statistics*, 26(4):873–882, 2017.

[132] B. M. Turner and P. B. Sederberg. Approximate Bayesian computation with differential evolution. *Journal of Mathematical Psychology*, 56(5):375–385, 2012.

[133] B. M. Turner and T. Van Zandt. A tutorial on approximate Bayesian computation. *Journal of Mathematical Psychology*, 56(2):69–85, 2012.

[134] B. M. Turner and T. Van Zandt. Hierarchical approximate Bayesian computation. *Psychometrika*, 79(2):185–209, 2014.

[135] M. K. Vakilzadeh, Y. Huang, J. L. Beck, and T. Abrahamsson. Approximate Bayesian Computation by Subset Simulation using hierarchical state-space models. *Mechanical Systems and Signal Processing*, 84:2–20, 2017.

[136] M. K. Vakilzadeh, J. L. Beck, and T. Abrahamsson. Using approximate Bayesian computation by subset simulation for efficient posterior assessment of dynamic state-space model classes. *SIAM Journal on Scientific Computing*, 40(1):B168–B195, 2018.

[137] A. Valor, F. Caleyo, L. Alfonso, D. Rivas, and J. Hallen. Stochastic modeling of pitting corrosion: a new model for initiation and growth of multiple corrosion pits. *Corrosion science*, 49(2):559–579, 2007.

[138] J. Van Noortwijk. A survey of the application of gamma processes in maintenance. *Reliability Engineering & System Safety*, 94(1):2–21, 2009.

[139] J. Van Noortwijk and H. Klatter. Optimal inspection decisions for the block mats of the Eastern-Scheldt barrier. *Reliability Engineering & System Safety*, 65 (3):203–211, 1999.

[140] J. Wakefield. *Bayesian and frequentist regression methods*. Springer Science & Business Media, 2013.

[141] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye. *Probability & statistics for engineers & scientists (9th edition)*. Boston: Prentice Hall, 2012.

[142] Y. Wang, H. Huang, L. Huang, and B. Ristic. Evaluation of Bayesian source estimation methods with Prairie Grass observations and Gaussian plume model: A comparison of likelihood functions and distance measures. *Atmospheric environment*, 152:519–530, 2017.

[143] Z. Wang, M. Broccardo, and J. Song. Hamiltonian Monte Carlo methods for Subset Simulation in reliability analysis. *Structural Safety*, 76:51–67, 2019.

[144] A. Weyant, C. Schafer, and W. M. Wood-Vasey. Likelihood-free cosmological inference with type Ia supernovae: approximate Bayesian computation for a complete treatment of uncertainty. *The Astrophysical Journal*, 764(2):116, 2013.

[145] R. D. Wilkinson. Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical applications in genetics and molecular biology*, 12(2):129–141, 2013.

[146] X.-X. Yuan. *Stochastic modeling of deterioration in nuclear power plant components*. PhD thesis, University of Waterloo, Waterloo, Canada, 2007. URL http://hdl.handle.net/10012/2756.

[147] X.-X. Yuan, M. Pandey, and G. Bickel. A probabilistic model of wall thinning in CANDU feeders due to flow-accelerated corrosion. *Nuclear Engineering and Design*, 238(1):16–24, 2008.

[148] X.-X. Yuan, D. Mao, and M. Pandey. A Bayesian approach to modeling and predicting pitting flaws in steam generator tubes. *Reliability Engineering & System Safety*, 94(11):1838–1847, 2009.

[149] S. Zhang and W. Zhou. Bayesian dynamic linear model for growth of corrosion

defects on energy pipelines. *Reliability Engineering & System Safety*, 128:24–31, 2014.

APPENDICES

Appendix A

Derivations of Joint Distributions

A.1 Joint Distribution of the Inspection Error

To formulate the joint PDF $f_{\Delta\mathbf{Z}_i}(\Delta\mathbf{z}_i)$, an important step is the following linear transformation: $\Delta\mathbf{Z}_i = \mathbf{J}\mathbf{Z}_i$. Here, $\mathbf{Z}_i = \{Z_{i0}, Z_{i1}, \cdots, Z_{im_i}\}$, $\Delta\mathbf{Z}_i = \{Z_{i1} - Z_{i0}, Z_{i2} - Z_{i1}, \cdots, Z_{i,m_i-1} - Z_{im_i}\}^T$ and $\mathbf{J}$ is the transformation matrix. The expression of $\mathbf{J}$ can be written as [79]

$$J = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix}_{m_i \times (m_i+1)} \tag{A.1}$$

Here, $\mathbf{Z}_i$ is a multivariate normal random vector with zero mean vector and covariance matrix $\mathbf{\Sigma}_{\mathbf{Z}_i} = \sigma_Z \mathbf{I}_{m_i+1}$, where $\mathbf{I}_{m_i+1}$ is an identity matrix of dimension (m_i+1). Hence, the transformed random vector $\Delta\mathbf{Z}_i$ will have a zero mean vector and a covariance matrix $\mathbf{\Sigma}_{\Delta\mathbf{Z}_i} = \sigma_Z^2 \mathbf{J}\mathbf{I}_{m_i+1}\mathbf{J}^T = \sigma_Z^2 \mathbf{J}\mathbf{J}^T$. Thus, the joint PDF $f_{\Delta\mathbf{Z}_i}(\Delta\mathbf{z}_i)$ can be written as

$$f_{\Delta\mathbf{Z}_i}(\Delta\mathbf{z}_i) = \frac{1}{(2\pi)^{m_i/2}|\mathbf{\Sigma}_{\Delta\mathbf{Z}_i}|} \exp\left\{ -\frac{1}{2}\Delta\mathbf{z}_i \mathbf{\Sigma}_{\Delta\mathbf{Z}_i}^{-1} \Delta\mathbf{z}_i^T \right\} \tag{A.2}$$

where $|\bullet|$ represents the determinant operator.

A.2 Joint Distribution of Flaw Sizes

The PDF of the measured flaw size $H_{ij}^{(m)}$ is the convolution of the PDF of the depth of detected flaws $H_{ij}^{(d)}$ and the measurement error Z as they are related by $H_{ij}^{(m)} = H_{ij}^{(d)} + Z$. Therefore,

$$f_{H_{ij}^{(m)}}(h_{ij}^{(m)}) = \int_0^\infty f_{H_{ij}^{(d)}}(y) f_Z(h_{ij}^{(m)} - y) dy \tag{A.3}$$

where $f_Z(\bullet)$ is the distribution of the measurement noise. The PDF of the detected flaw size $H_{ij}^{(d)}$ is a conditional probability, which is expressed as

$$f_{H_{ij}^{(d)}}(h) = f_H(h \mid D = 1) = \frac{f(D = 1 \mid H = h) f_H(h)}{\mathbb{P}[D = 1]} \tag{A.4}$$

The first term of the numerator is just the POD function, as defined in Equation 2.26, and the second term of the numerator is the Weibull distribution assumed for the actual pit depth in Equation 2.25. The denominator is the unconditional flaw size detection probability, which can be expressed as

$$\mathbb{P}[D = 1] = \mathbb{E}[p(h)] = \int_0^\infty p(h) f_H(h) dh \tag{A.5}$$

Substituting Equation A.5 and Equation A.4 back into Equation A.3, we obtain

$$f_{H_{ij}^{(m)}}(h_{ij}^{(m)}) = \frac{1}{\mathbb{E}[p(h)]} \int_0^\infty p(h) f_H(h) f_Z(h_{ij}^{(m)} - h) dh \tag{A.6}$$

The joint distribution of the flaw sizes now can be written as

$$f_{\mathbf{H}_i^{(m)}}(\mathbf{h}_i^{(m)}) = \prod_{j=1}^{n_{di}} f_{H_{ij}^{(m)}}(h_{ij}^{(m)}) = \prod_{j=1}^{n_{di}} \left\{ \frac{1}{\mathbb{E}[p(h)]} \int_0^\infty p(h) f_H(h) f_Z(h_{ij}^{(m)} - h) dh \right\} \quad \text{(A.7)}$$

Equation A.7 is same as Equation 2.28.

A.3 Probability of the Number of Detected Flaws

The probability of the number of detected flaws given the number of newly generated flaws, i.e., $\mathbb{P}[N_{di} = n_{di} \mid N_i = n_i]$, can be calculated from a binomial distribution with the success probability equal to $\mathbb{E}[p(h)]$. Therefore,

$$\mathbb{P}[N_{di} = n_{di} \mid N_i = n_i] = \frac{n_i!}{(n_i - n_{di})! n_{di}!} \{\mathbb{E}[p(h)]\}^{n_{di}} \{1 - \mathbb{E}[p(h)]\}^{n_i - n_{di}} \quad \text{(A.8)}$$

Finally, the probability of the number of detected flaws, i.e., $\mathbb{P}[N_{di} = n_{di}]$, can be calculated as,

$$\begin{aligned} \mathbb{P}[N_{di} = n_{di}] &= \sum_{n_i=0}^{\infty} \mathbb{P}[N_{di} = n_{di} \mid N_i = n_i] \mathbb{P}[N_i = n_i] \\ &= \sum_{n_i=0}^{\infty} \frac{\{\Lambda(t_i, t_{i-1})\}^{n_i}}{(n_i - n_{di})! n_{di}!} \exp\{-\Lambda(t_i, t_{i-1})\} \{\mathbb{E}[p(h)]\}^{n_{di}} \{1 - \mathbb{E}[p(h)]\}^{n_i - n_{di}} \end{aligned} \quad \text{(A.9)}$$

Equation A.9 is same as Equation 2.30.

Appendix B

Markov Chain: Basic Concepts

A Markov process $\mathbf{X}(t)$ follows the Markov property, according to which the outcome of a stochastic process at any instant of time is dependent only on its immediately preceding outcome [101]. In other words, the process $\mathbf{X}(t)$ is not influenced by the past, but only by the present, i.e.,

$$\mathbb{P}[\mathbf{X}(t_n) \leq \mathbf{x}_n \mid \mathbf{X}(t), t \leq t_{n-1} < t_n] = \mathbb{P}[\mathbf{X}(t_n) \leq \mathbf{x}_n \mid \mathbf{X}(t_{n-1})] \tag{B.1}$$

In the case of a discrete-time Markov process, if $t_1 < t_2 < \cdots < t_{n-1} < t_n < \cdots$, then

$$\mathbb{P}[\mathbf{X}(t_n) \leq \mathbf{x}_n \mid \mathbf{X}(t_{n-1}), \mathbf{X}(t_{n-2}), \cdots, \mathbf{X}(t_1)] = \mathbb{P}[\mathbf{X}(t_n) \leq \mathbf{x}_n \mid \mathbf{X}(t_{n-1})] \tag{B.2}$$

Markov chains are a special kind of Markov process where $\mathbf{X}(t)$ undergoes transitions from one state to another between a set of finite or countably infinite states $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_j, \cdots$, on a state space $\mathcal{S}$. Markov chains can be discrete-time or continuous-time; however, in the context of MCMC, Markov chains are considered to be discrete-time processes. A. A. Markov first introduced the Markov chain theory for a finite state space, although Kolmogorov was the one who proposed the theory of Markov

chains on a countably infinite state space [101].

Let us assume that $\mathbf{X}_n = \mathbf{X}(t_n)$ is the state of the system at $t = t_n$ and $n \geq m \geq 0$. Now, the probability that the process $\mathbf{X}(t)$ occupies the state $\mathbf{s}_j$ at time $t = t_n$, given that it was in state $\mathbf{s}_i$ at time $t = t_m$, is represented by the transition probability $p_{ij}(m, n)$, which can be written as

$$p_{ij}(m, n) = \mathbb{P}[\mathbf{X}_n = \mathbf{s}_j \mid \mathbf{X}_m = \mathbf{s}_i] \tag{B.3}$$

The transition probabilities can be arranged in a matrix form as

$$P(m, n) = \begin{pmatrix} p_{11}(m, n) & p_{12}(m, n) & \cdots & p_{1j}(m, n) & \cdots \\ p_{21}(m, n) & p_{22}(m, n) & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_{i1}(m, n) & \cdots & \cdots & p_{ij}(m, n) & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \tag{B.4}$$

where $P(m, n)$ is called the transition probability matrix. The matrix elements are non-negative, and the elements of each row sum to unity, i.e., $\sum_j p_{ij}(m, n) = 1$. A Markov chain is completely defined by its transition probability matrix $P(m, n)$ and its initial probability, $p_r(1) = \mathbb{P}[\mathbf{X}_1 = \mathbf{s}_r]$.

B.1 Time-Homogeneous Markov Chain

In the context of MCMC, the transition probabilities need to be stationary [101] – a property of the time-homogeneous Markov chain. In other words, a Markov chain is time-homogeneous if the transition probability is only dependent on the time-difference,

i.e.,

$$\mathbb{P}[\mathbf{X}_n = \mathbf{s}_j \mid \mathbf{X}_m = \mathbf{s}_i] = p_{ij}(n-m) = p_{ij}^{(k)} \tag{B.5}$$

where $p_{ij}^{(k)}$ is the probability that a Markov chain undergoes transition from state $\mathbf{s}_i$ to state $\mathbf{s}_j$ in k steps. Thus, $P^{(k)} = \{p_{ij}^{(k)}\}$ is the k step transition probability matrix. For $k = 1$, $p_{ij}^{(1)}$ can be simply denoted as p_{ij}. Thus, the one-step transition probability matrix for a time-homogeneous Markov chain can be written as

$$P^{(1)} = P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1j} & \cdots \\ p_{21} & p_{22} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_{i1} & \cdots & \cdots & p_{ij} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \tag{B.6}$$

B.2 Chapman-Kolmogorov Equation

The transition probability function follows the Chapman-Kolmogorov equation [101], i.e., for any $n > r > m$, we have

$$p_{ij}(m,n) = \sum_k p_{ik}(m,r) p_{kj}(r,n) \tag{B.7}$$

Accordingly, the probability transition matrices are related by the following expression

$$P(m,n) = P(m,r)P(r,n) \tag{B.8}$$

For a homogeneous Markov chain, the relation can be written as

$$p_{ij}^{(m+n)} = \sum_k p_{ik}^{(m)} p_{kj}^{(n)} = \sum_k p_{ik}^{(n)} p_{kj}^{(m)} \tag{B.9}$$

In matrix form, the same relation is expressed as

$$P^{(m+n)} = P^{(m)}P^{(n)} = P^{(n)}P^{(m)} \tag{B.10}$$

From Equation B.10, the one-step recursion relation can be derived as

$$P^{(n+1)} = P^{(n)}P^{(1)} = P^{(1)}P^{(n)}, \;\; n = 1, 2, 3, \cdots \tag{B.11}$$

Thus, an n step transition probability matrix can be easily derived by multiplying the one-step transition probability matrix n times.

Example: Random Walk

As an example of a Markov chain, a general one-dimensional random walk can be considered [101]. It is a Markov chain with possible states, $\mathbf{s}_1, \mathbf{s}_2, \cdots$, where the probability of transitioning from state $\mathbf{s}_j$ to $\mathbf{s}_{j+1}$ is p_j, to $\mathbf{s}_{j-1}$ is q_j, and the probability of remaining at the same state is r_j. For state $\mathbf{s}_1$, the process can stay there with probability r_1 or go to the next state $\mathbf{s}_2$ with probability p_1. A schematic of the general one-dimensional random walk is presented in Figure B.1. Accordingly,

$$\begin{aligned} &r_1 + p_1 = 1 \\ &q_j + r_j + p_j = 1, \;\; j = 2, 3, \cdots \end{aligned} \tag{B.12}$$

Consequently, the transition probability matrix P can be written as

$$P = \begin{pmatrix} r_1 & p_1 & 0 & 0 & 0 & \cdots \\ q_2 & r_2 & p_2 & 0 & 0 & \cdots \\ 0 & q_3 & r_3 & p_3 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{pmatrix} \tag{B.13}$$

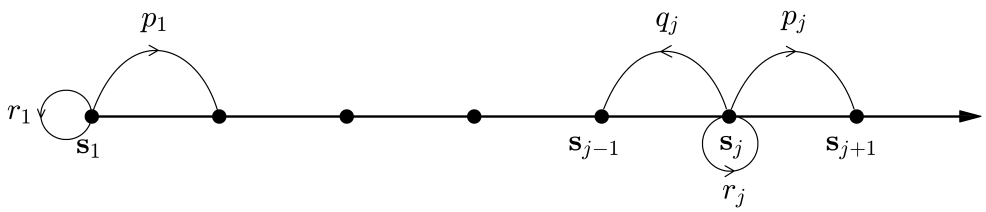


Figure B.1: One-dimensional random walk [101].

Depending on the element properties of the transition probability matrix, different kinds of random walks can be derived. Table B.1 contains some variants of random walks along with their properties. Note that the variants of random walks contain a finite number of states, $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_N$.

B.3 Stationary Distributions

Suppose a homogeneous Markov chain has a finite number of states, $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_N$. Let us say, at $t = t_n$, the probability vector for the chain location $\{\mathbf{X}_n = \mathbf{s}_j\}$ is given by $\Pi^{(n)} = \{\pi_1^{(n)}, \pi_2^{(n)}, \cdots, \pi_N^{(n)}\}$, $n = 1, 2, , \cdots$, where $\sum_{i=1}^{N} \pi_i^{(n)} = 1$. Now, the probability vector for the next chain location $\mathbf{X}_{n+1}$ can be easily derived using the recursion equation

$$\Pi^{(n+1)} = \Pi^{(n)} P^{(1)} = \Pi^{(1)} P^{(n)}, \;\; n = 1, 2, 3, \cdots \tag{B.14}$$

Now, a Markov chain is said to be stationary if,

$$\Pi^{(n)} \longrightarrow \Pi^{*}, \;\; n \longrightarrow \infty \tag{B.15}$$

Table B.1: Variants of one-dimensional random walks [101] .

Random Walk	**Description**	**Transition Probability Matrix**
Random walk with absorbing barriers	From an interior state, transitioning to left and right is possible with probabilities q and p, respectively ($p+q=1$). However, no transition is possible from end states.	$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdot & \cdot & 0 \\ q & 0 & p & 0 & 0 & \cdot & \cdot & 0 \\ 0 & q & 0 & p & 0 & \cdot & \cdot & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdot & \cdot & \cdot & q & 0 & p \\ 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 & 1 \end{pmatrix}$
Random walk with reflecting barriers	If the process reaches a boundary, it reflects the process back to the adjacent state.	$P = \begin{pmatrix} q & p & 0 & 0 & 0 & \cdot & \cdot & 0 \\ q & 0 & p & 0 & 0 & \cdot & \cdot & 0 \\ 0 & q & 0 & p & 0 & \cdot & \cdot & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdot & \cdot & \cdot & q & 0 & p \\ 0 & 0 & \cdot & \cdot & \cdot & 0 & q & p \end{pmatrix}$
Cyclic random walk	The two end-boundary states, $\mathbf{s}_1$ and $\mathbf{s}_N$, connect together to form a circle. The random walk continues endlessly in this circular path.	$P = \begin{pmatrix} 0 & p & 0 & 0 & 0 & \cdot & \cdot & 0 & q \\ q & 0 & p & 0 & 0 & \cdot & \cdot & 0 & 0 \\ 0 & q & 0 & p & 0 & \cdot & \cdot & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdot & \cdot & \cdot & \cdot & q & 0 & p \\ p & 0 & \cdot & \cdot & \cdot & \cdot & 0 & q & 0 \end{pmatrix}$

irrespective of the initial distribution. In another way, regardless of the initial starting state, the transition probabilities converge to a limiting probability, i.e.,

$$p_{ij}^{(n)} \longrightarrow p_j^*, \;\; n \longrightarrow \infty \tag{B.16}$$

Since Π^* is an invariant distribution, multiplying it with $P^{(1)}$ will essentially give back the same distribution:

$$\begin{aligned} \Pi^* P^{(1)} &= \Pi^* \\ \implies \Pi^*(P^{(1)} - I) &= \mathbf{0} \end{aligned} \tag{B.17}$$

where I is an identity matrix. To numerically calculate the steady state vector, we can either solve this system of linear equations or keep on multiplying $P^{(1)}$ with $\Pi^{(1)}$ until n reaches a reasonable value and $\Pi^{(n)}$ converges to the steady state.

However, a Markov chain may not have a stationary distribution, or even if the chain does own one, it may not be unique. To guarantee the existence of a unique stationary distribution, the Markov chain has to follow two constraints:

1. **Irreducibility**: A Markov chain is said to be irreducible (a communicating chain) if any state $\mathbf{s}_j$ can be reached from any other state $\mathbf{s}_i$ with a non-zero probability, i.e.,

$$p_{ij}^{(n)} > 0, \;\; n = 1, 2, 3, \cdots \tag{B.18}$$

2. **Aperiodicity**: A Markov chain is said to be aperiodic if the process never returns to the same state with a fixed period. Mathematically, the process has a zero transition probability for returning to the same state in kT steps, where T is the fixed return period:

$$p_{jj}^{(n)} = 0, \;\; n = kT, \;\; k = 1, 2, \cdots \tag{B.19}$$

Any Markov chain that follows these two properties is said to be ergodic. An ergodic Markov chain is guaranteed to have a unique steady state probability vector Π^* [101].

B.4 Reversible Markov Chain

To ensure that any probability vector $\Pi^* = \{\pi_1^*, \pi_2^*, \cdots, \pi_N^*\}$ is in fact the desired steady state vector, the Markov chain has to follow the sufficient (but not necessary) reversibility condition [6]:

$$\begin{aligned} & \pi_i^* p_{ij} = \pi_j^* p_{ji} \\ \Longrightarrow\ & \pi_i^* \mathbb{P}[\mathbf{X}_{n+1} = \mathbf{s}_j \mid \mathbf{X}_n = \mathbf{s}_i] = \pi_j^* \mathbb{P}[\mathbf{X}_{n+1} = \mathbf{s}_i \mid \mathbf{X}_n = \mathbf{s}_j] \end{aligned} \tag{B.20}$$

where π_i^* and π_j^* are the equilibrium probabilities of being in states $\mathbf{s}_i$ and $\mathbf{s}_j$, respectively. This expression is also known as the detailed balance equation. Generally speaking, the detailed balance equation guarantees that in a long run, a time-reversible Markov chain spends equal amounts of time to move from state $\mathbf{s}_i$ to $\mathbf{s}_j$ and vice versa. For a time-reversible Markov chain, we can show that

$$\sum_i \pi_i^* p_{ij}^{(1)} = \sum_i \pi_j^* p_{ji}^{(1)} = \pi_j^* \sum_i p_{ji}^{(1)} = \pi_j^* \tag{B.21}$$

This states that for a time-reversible Markov chain, the amount of time spent at state $\mathbf{s}_j$ is equal to the total amount of time spent by the chain transitioning from other states, i.e., $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_j, \cdots, \mathbf{s}_N$ to $\mathbf{s}_j$. In other words, a time reversible Markov chain does not have a net flow of probability through its closed cycle of states. For example,

$$p_{ij} p_{jk} p_{ki} = p_{ik} p_{kj} p_{ji}, \quad \forall i, j, k \tag{B.22}$$

This is known as the Kolmogorov's criterion – a necessary and sufficient condition for the reversibility condition of Markov chains. If we design a Markov chain such that it follows the detailed balance equation, then it will have Π^* as its stationary distribution. Thus, the detailed balance equation is often used to derive various MCMC samplers.

In the case of a continuous state-space, the reversibility condition or detailed balance equation can be easily extended as

$$\pi(\mathbf{s})\mathcal{K}(\mathbf{s}' \mid \mathbf{s}) = \pi(\mathbf{s}')\mathcal{K}(\mathbf{s} \mid \mathbf{s}') \tag{B.23}$$

where the steady state vector Π^* becomes a stationary probability density $\pi(\bullet)$ and the one-step transition matrix $P^{(1)} = P$ becomes the transition kernel probability density $\mathcal{K}(\mathbf{s}' \mid \mathbf{s})$ which represents a transition of the chain from state $\mathbf{s}$ to $\mathbf{s}'$.

Example

Suppose a construction company needs to buy building materials for a construction project. The company has two choices: (i) materials supplier A, and (ii) materials supplier B. The probability of choosing a materials supplier, given the choice of the previous construction project, can be represented by a transition probability matrix:

$$P = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix} \tag{B.24}$$

The matrix P represents that the company is 70% likely to give a contract to supplier A and 30% likely to give the same contract to supplier B, given that it hired supplier A in a previous construction project. Similarly, there is 60% probability that the company will select supplier B, and 40% for supplier A if it had chosen B in the previous project. Figure B.2 shows a graphical representation of the transition probabilities.

Suppose the company hired supplier A in its previous project. This can be represented as

$$\Pi^{(1)} \begin{pmatrix} 1 & 0 \end{pmatrix} \tag{B.25}$$

which says supplier A has 100% probability of selection and supplier B has 0%. Using

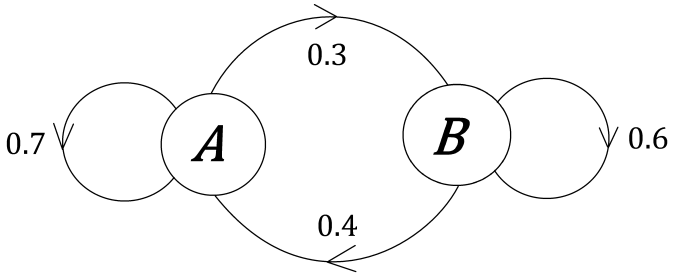


Figure B.2: A graphical representation of the transition probability matrix.

Equation B.14, the choice can be predicted for the next project as

$$\Pi^{(2)} = \Pi^{(1)}P = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix} = \begin{pmatrix} 0.7 & 0.3 \end{pmatrix} \tag{B.26}$$

Similarly, predictions can also be made for the 3rd, 4th, 5th, $\cdots$ projects as

$$\Pi^{(3)} = \Pi^{(2)}P = \begin{pmatrix} 0.61 & 0.39 \end{pmatrix} \qquad \Pi^{(9)} = \Pi^{(8)}P = \begin{pmatrix} 0.5715 & 0.4285 \end{pmatrix}$$

$$\Pi^{(4)} = \Pi^{(3)}P = \begin{pmatrix} 0.583 & 0.417 \end{pmatrix} \qquad \Pi^{(10)} = \Pi^{(9)}P = \begin{pmatrix} 0.5714 & 0.4286 \end{pmatrix}$$

$$\Pi^{(5)} = \Pi^{(4)}P = \begin{pmatrix} 0.5749 & 0.4251 \end{pmatrix} \qquad \Pi^{(11)} = \Pi^{(10)}P = \begin{pmatrix} 0.5714 & 0.4286 \end{pmatrix}$$

$$\Pi^{(6)} = \Pi^{(5)}P = \begin{pmatrix} 0.5725 & 0.4275 \end{pmatrix} \qquad \Pi^{(12)} = \Pi^{(11)}P = \begin{pmatrix} 0.5714 & 0.4286 \end{pmatrix}$$

$$\Pi^{(7)} = \Pi^{(6)}P = \begin{pmatrix} 0.5717 & 0.4283 \end{pmatrix} \qquad \Pi^{(13)} = \Pi^{(12)}P = \begin{pmatrix} 0.5714 & 0.4286 \end{pmatrix}$$

$$\Pi^{(8)} = \Pi^{(7)}P = \begin{pmatrix} 0.5715 & 0.4285 \end{pmatrix} \qquad \Pi^{(14)} = \Pi^{(12)}P = \begin{pmatrix} 0.5714 & 0.4286 \end{pmatrix}$$

It can be observed that the probability vector reached a steady state condition after the 10th update. This implies that the probabilities of selecting different materials suppliers converge to limiting values. To prove that the steady state distribution is the same or invariant irrespective of the initial distribution, let us start with an arbitrary

initial probability vector:

$$\Pi^{(1)} = \begin{pmatrix} 0.45 & 0.55 \end{pmatrix} \qquad \Pi^{(6)} = \Pi^{(5)}P = \begin{pmatrix} 0.5711 & 0.4289 \end{pmatrix}$$

$$\Pi^{(2)} = \Pi^{(1)}P = \begin{pmatrix} 0.535 & 0.465 \end{pmatrix} \qquad \Pi^{(7)} = \Pi^{(6)}P = \begin{pmatrix} 0.5713 & 0.4287 \end{pmatrix}$$

$$\Pi^{(3)} = \Pi^{(2)}P = \begin{pmatrix} 0.5605 & 0.4395 \end{pmatrix} \qquad \Pi^{(8)} = \Pi^{(7)}P = \begin{pmatrix} 0.5714 & 0.4286 \end{pmatrix}$$

$$\Pi^{(4)} = \Pi^{(3)}P = \begin{pmatrix} 0.5681 & 0.4318 \end{pmatrix} \qquad \Pi^{(9)} = \Pi^{(8)}P = \begin{pmatrix} 0.5714 & 0.4286 \end{pmatrix}$$

$$\Pi^{(5)} = \Pi^{(4)}P = \begin{pmatrix} 0.5704 & 0.4296 \end{pmatrix} \qquad \Pi^{(10)} = \Pi^{(9)}P = \begin{pmatrix} 0.5714 & 0.4286 \end{pmatrix}$$

Notice that the steady state vector $\Pi^* = \begin{pmatrix} 0.5714 & 0.4286 \end{pmatrix}$ is invariant irrespective of the initial conditions. These limiting probabilities can also be calculated by multiplying the transition probability matrix by itself several times until it converges to a steady state condition:

$$P = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix} \qquad P^6 = \begin{pmatrix} 0.5717 & 0.4283 \\ 0.5710 & 0.4290 \end{pmatrix}$$

$$P^2 = \begin{pmatrix} 0.61 & 0.39 \\ 0.52 & 0.48 \end{pmatrix} \qquad P^7 = \begin{pmatrix} 0.5715 & 0.4285 \\ 0.5713 & 0.4287 \end{pmatrix}$$

$$P^3 = \begin{pmatrix} 0.583 & 0.417 \\ 0.556 & 0.444 \end{pmatrix} \qquad P^8 = \begin{pmatrix} 0.5715 & 0.4285 \\ 0.5714 & 0.4286 \end{pmatrix}$$

$$P^4 = \begin{pmatrix} 0.5749 & 0.4251 \\ 0.5668 & 0.4332 \end{pmatrix} \qquad P^9 = \begin{pmatrix} 0.5714 & 0.4286 \\ 0.5714 & 0.4286 \end{pmatrix}$$

$$P^5 = \begin{pmatrix} 0.5725 & 0.4275 \\ 0.5700 & 0.4300 \end{pmatrix} \qquad P^{10} = \begin{pmatrix} 0.5714 & 0.4286 \\ 0.5714 & 0.4286 \end{pmatrix}$$

However, the easiest approach for calculating the stationary distribution is by using Equation B.17:

$$
\begin{aligned}
&\Pi^*(P - I) = \mathbf{0} \\
\implies &\begin{pmatrix} \pi_1^* & \pi_2^* \end{pmatrix} \left[\begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right] = \begin{pmatrix} 0 & 0 \end{pmatrix} \\
\implies &\begin{pmatrix} \pi_1^* & \pi_2^* \end{pmatrix} \begin{pmatrix} -0.3 & 0.3 \\ 0.4 & -0.4 \end{pmatrix} = \begin{pmatrix} 0 & 0 \end{pmatrix} \\
\implies &0.3\pi_1^* - 0.4\pi_2^* = 0
\end{aligned}
\tag{B.27}
$$

Since Π^* is a probability distribution, we know that

$$
\pi_1^* + \pi_2^* = 1 \tag{B.28}
$$

Solving this pair of simultaneous equations (Equation B.27 and Equation B.28) gives the steady state distribution as

$$
\Pi^* = \begin{pmatrix} \pi_1^* & \pi_2^* \end{pmatrix} = \begin{pmatrix} 4/7 & 3/7 \end{pmatrix} = \begin{pmatrix} 0.5714 & 0.4286 \end{pmatrix} \tag{B.29}
$$

Hence, in the long run, there is a 57.14% chance that suppler A will get hired by the construction company, whereas, for suppler B, the probability is only 42.86%.

Appendix C

Modified Metropolis Algorithm

The steps for implementing the MM algorithm is described in this appendix. At any simulation level s, $s = 1, 2, \cdots, S$, the MM algorithm starts by identifying the independent components of the model parameters such as $\mathbf{\Theta}^{(k)} = \{\Theta_1^{(k)}, \Theta_2^{(k)}, \cdots, \Theta_{c_0}^{(k)}\}$, where $\Theta_c^{(k)}$, $c = 1, 2, \cdots, c_0$, can be a scalar or a vector of parameters. Transforming the joint prior distribution into a product of independent prior distributions: $f(\mathbf{\Theta}) = f_1(\Theta_1) f_2(\Theta_2) \cdots f_{c_0}(\Theta_{c_0})$, the MM acceptance probabilities for each component can be calculated as

$$\mathcal{A}_c = \min\left\{1, \frac{f_c(\tilde{\Theta}_c) q_c(\Theta_c^{(k)} \mid \tilde{\Theta}_c)}{f_c(\Theta_c^{(k)}) q_c(\tilde{\Theta}_c \mid \Theta_c^{(k)})}\right\} \tag{C.1}$$

where $\tilde{\Theta}_c \sim q_c(\Theta \mid \Theta_c^{(k)})$ and $q_c(\Theta \mid \Theta_c^{(k)})$ is the cth component's proposal distribution conditioned on the current sample $\Theta_c^{(k)}$. To propose a new parameter sample $\mathbf{\Theta}^* = \{\Theta_1^*, \Theta_2^*, \cdots, \Theta_{c_0}^*\}$, each component of it is updated based on their respective MM acceptance probabilities $\mathcal{A}_c$, i.e., $\Theta_c^* = \tilde{\Theta}_c$ is set with probability $\mathcal{A}_c$ and $\Theta_c^* = \Theta_c^{(k)}$ is set with probability $1 - \mathcal{A}_c$. Once the proposed parameter $\mathbf{\Theta}^*$ is obtained, a data set $\mathbf{D}^*$ is simulated from the forward model $\mathcal{M}(\mathbf{D} \mid \mathbf{\Theta}^*)$ and the distance function

$\rho^* = \rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}})$ is evaluated. Finally, the model parameter is updated based on the tolerance threshold $\epsilon^{(s-1)}$, i.e., $\boldsymbol{\Theta}^{(k+1)} = \boldsymbol{\Theta}^*$, if $\rho^* \leq \epsilon^{(s-1)}$; otherwise $\boldsymbol{\Theta}^{(k+1)} = \boldsymbol{\Theta}^{(k)}$.

The pseudocode of the MM sampler is presented in Algorithm 14. The component-specific proposal distribution $q_c(\Theta \mid \Theta_c^{(k)})$ is assumed to be a normal density function with a mean of $\Theta_c^{(k)}$ and standard deviation σ_c (or covariance matrix $\boldsymbol{\Sigma}_c$) so that $q_c(\Theta \mid \Theta_c^{(k)}) = q_c(\Theta_c^{(k)} \mid \Theta)$. The parameter σ_c (or covariance matrix $\boldsymbol{\Sigma}_c$) can be adaptively obtained at each simulation level by equating it to the sample standard deviation (or covariance matrix) of the $n_0 p_0$ number of selected parameter samples.

Algorithm 14 MM sampler for ABC-SS

1: **for** $c = 1$ to c_0 **do**
2: Generate $\tilde{\Theta}_c \sim q_c(\Theta \mid \Theta_c^{(s,h+r-1)})$, where $\boldsymbol{\Theta}^{(s,k)} = \{\Theta_1^{(s,k)}, \cdots, \Theta_{c_0}^{(s,k)}\}$
3: Assuming $f(\boldsymbol{\Theta}) = f_1(\Theta_1) f_2(\Theta_2) \cdots f_{c_0}(\Theta_{c_0})$, calculate

$$\mathcal{A}_c = \min\left\{1, \frac{f_c(\tilde{\Theta}_c) q_c(\Theta_c^{(s,h+r-1)} \mid \tilde{\Theta}_c)}{f_c(\Theta_c^{(s,h+r-1)}) q_c(\tilde{\Theta}_c \mid \Theta_c^{(s,h+r-1)})}\right\}$$

4: Generate $u \sim \mathcal{U}[0,1]$
5: Set

$$\Theta_c^* = \begin{cases} \tilde{\Theta}_c, & \text{if } u \leq \mathcal{A}_c \\ \Theta_c^{(s,h+r-1)}, & \text{otherwise} \end{cases}$$

6: **end for**
7: Simulate $\mathbf{D}^* \sim \mathcal{M}(\mathbf{D} \mid \boldsymbol{\Theta}^*)$
8: Evaluate $\rho^* = \rho(\mathbf{D}^*, \mathbf{D}_{\text{obs}})$
9: Set

$$\{\boldsymbol{\Theta}^{(s,h+r)}, \rho^{(s,h+r)}\} = \begin{cases} \{\boldsymbol{\Theta}^*, \rho^*\}, & \text{if } \rho^* \leq \epsilon^{(s-1)} \\ \{\boldsymbol{\Theta}^{(s,h+r-1)}, \rho^{(s,h+r-1)}\}, & \text{otherwise} \end{cases}$$

Appendix D

Simulation Examples using Gamma Process

Three examples are presented to illustrate the application of the proposed ABC-MCMC method. The first illustrates the convergence behavior of the ABC posterior, whereas the second demonstrates the effectiveness of the method in capturing the initial degradation and the non-stationary trend of the degradation process. The third example shows the applicability of the method in estimating the lifetime distribution of a component population. In all three examples, the Bayesian inference schemes are implemented in MATLAB® 2017b on a desktop computer with Intel i5-6500 processor.

D.1 Example 1: A One-Parameter Stationary Model

Assuming zero initial degradation, synthetic data consisting measurements related to degradation of five fictitious components from three repeated inspections, taken at 5, 10, and 15 years, are simulated from a stationary gamma process with parameters $\alpha = 4$, $\eta = 1$, and $\beta = 0.015$. The normally distributed sizing error has zero mean

and standard deviation $\sigma_Z = 0.1$ mm. Assuming α to be the only unknown parameter, both ABC-MCMC and L-MCMC methods are implemented to estimate it. The prior distribution of α is selected to be a uniform distribution, i.e., $f(\alpha) = \mathcal{U}[0, 10]$. The proposal distribution is chosen to be a normal distribution with mean equal to the current sample and COV equal to 0.1 for both methods. In the L-MCMC method, the likelihood function is numerically integrated using Monte Carlo simulation [76] with 500 samples.

For implementing the ABC-MCMC scheme, four values of the tolerance threshold are considered, i.e., $\epsilon = \{0.6, 0.5, 0.4, 0.27\}$. These gradually decreasing tolerance thresholds help to observe the convergence of the ABC posterior. For initializing the ABC-MCMC run, Algorithm 13 is employed. The target threshold ϵ is reached through the following sequence of threshold values: $2\epsilon > 1.75\epsilon > 1.5\epsilon > 1.25\epsilon > \epsilon$. Selection of a thinning interval is performed by plotting sample ACF plots of the marginal Markov chains [22].

ABC-MCMC generated three chains for each case as required by the criterion of GR statistic. For the purpose of representation, Figure D.1 shows the three MCMC runs and the corresponding sample ACF and GR statistic plots for the $\epsilon = 0.27$ case. Accordingly, the chosen chain lengths and thinning intervals along with the computation times are presented in Table D.1; the computation times are given for single MCMC runs. The sample ACF plots of $\epsilon = 0.27$ case (Figure D.1) show very high correlation among the samples, and that is why a larger thinning interval of 30000 is chosen. Although the required length of the MCMC run given by the GR statistic is only around 2.5×10^6, given the high amount of thinning, a length of 10^7 is selected. The corresponding 0.01th percentile values of the distributions of the proposed distance function values (calculated using the proposed samples from all three runs of the algorithm) are 0.282 ($\epsilon = 0.6$), 0.278 ($\epsilon = 0.5$), 0.276 ($\epsilon = 0.4$) and 0.274 ($\epsilon = 0.27$), respectively. It

can be noticed that the first three tolerance thresholds, $\epsilon = \{0.6, 0.5, 0.4\}$, are far above the threshold choice criterion $Q_\epsilon \leq 0.01$, whereas $\epsilon = 0.27 < 0.274$ meets the criterion proposed for tolerance selection. Figure D.2 shows the distributions of the proposed distance values and the regions of the corresponding chosen tolerance thresholds.

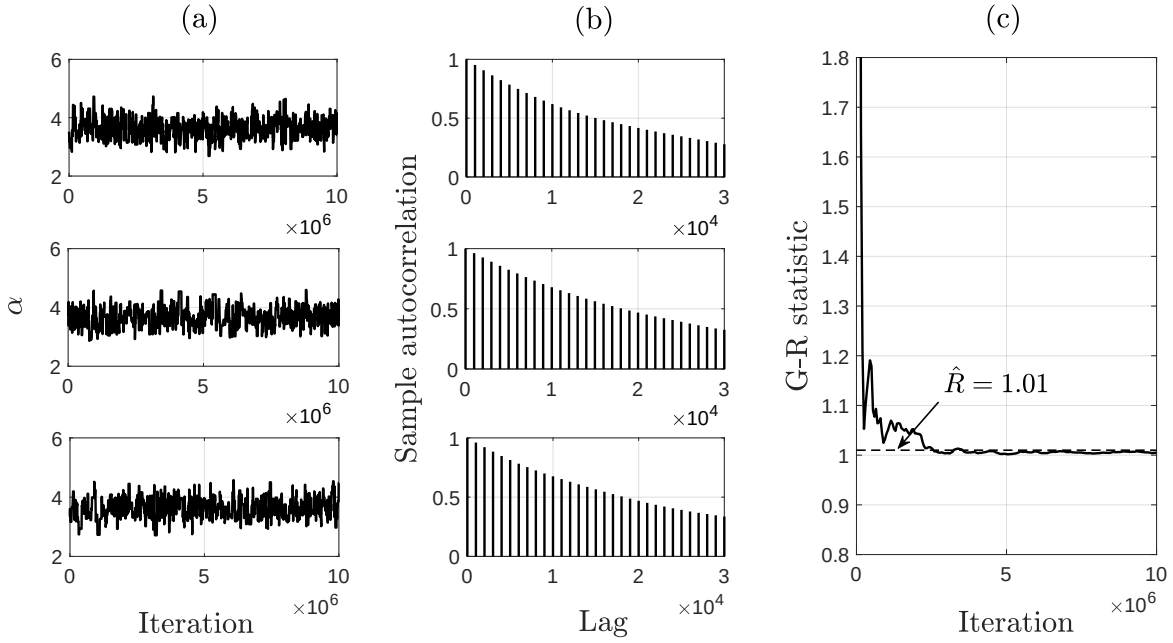


Figure D.1: (a) Three MCMC runs of the parameter α generated by ABC-MCMC with $\epsilon = 0.27$, and the corresponding (b) sample ACF plots, and (c) the convergence of the GR statistic.

Table D.1: Selected attributes of the MCMC chains and computation times.

Method	Tolerance (ϵ)	Chain length	Burn-in length	Thinning interval	Computation time
ABC-MCMC	0.6	1×10^5	–	100	3.3 s
	0.5	2×10^5	–	200	6.4 s
	0.4	2×10^5	–	500	6.4 s
	0.27	1×10^7	–	30000	320 s
L-MCMC	–	1×10^3	100	3	860 s

Similarly, the L-MCMC chain attributes and its computation time are presented in Table D.1. Figure D.3 shows three MCMC runs and the corresponding plots for sample ACFs and GR statistic. As it can be observed that the sample autocorrelation

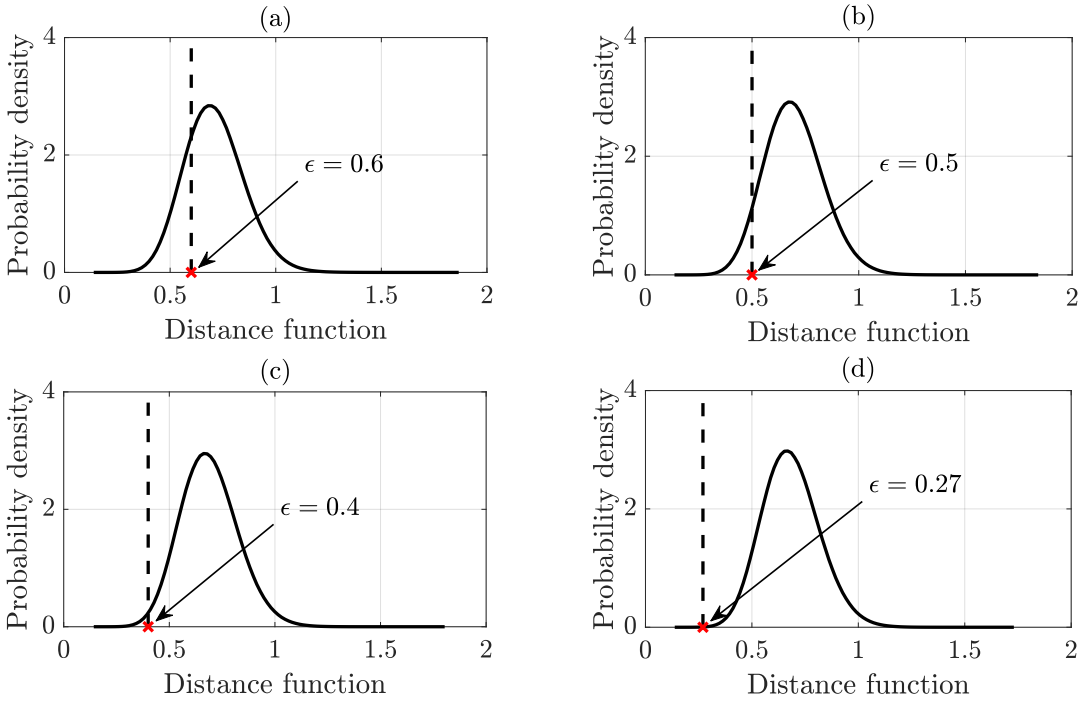


Figure D.2: Distributions of the proposed distance values for (a) $\epsilon = 0.6$*, (b)* $\epsilon = 0.5$*, (c)* $\epsilon = 0.4$*, and (d)* $\epsilon = 0.27$ *cases. Tolerance threshold* $\epsilon = 0.27$ *satisfies the tolerance selection criterion.*

of the L-MCMC output is much less compared to the likelihood-free approach. The GR statistic indicates a convergence at around 750 iteration length after burn-in; hence, an MCMC run of 1000 iterations is selected, and the initial 100 samples are discarded to allow for burn-in.

Figure D.4 shows the posterior distributions of α produced by both methods using samples from three MCMC runs after burn-in and thinning. The convergence of the ABC posterior is clearly visible as the tolerance threshold gradually decreases. The ABC posterior with $\epsilon = 0.27$ shows a very good match to the posterior generated using the likelihood-based method. The mean and COV of the ABC posterior with $\epsilon = 0.27$ are 3.65 and 0.096, respectively; whereas the posterior estimated by the likelihood-based method has a mean of 3.66 and COV of 0.078. This proves that ABC-MCMC

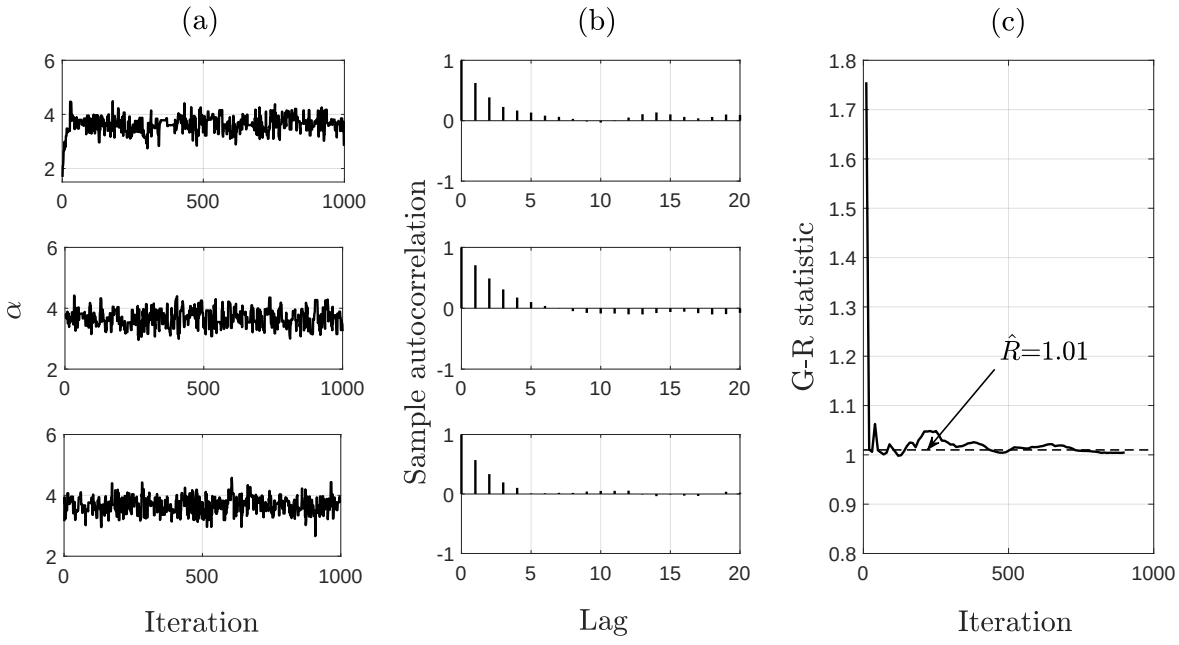


Figure D.3: (a) Three MCMC runs of the parameter α generated by L-MCMC, and the corresponding (b) sample ACF plots, and (c) the convergence of the GR statistic.

can accurately estimate the model parameters and it is computationally more efficient than the standard L-MCMC method. In fact, we were able to get around 170% more efficiency using the proposed approach than the traditional likelihood-based approach.

D.2 Example 2: A Two-Parameter Non-Stationary Model

Degradation paths of ten fictitious components are simulated, and a synthetic measurement data set is generated based on the following parameters: $\alpha = 2$, $\eta = 2.5$, $\beta = 0.01$, $\mu_A = 0.5$ mm, $\sigma_A = 0.1$ mm, and $\sigma_Z = 0.1$ mm. The observations are simulated for three repeated measurements at 2nd, 4th and 6th years of operation. Let us assume that the parameters η and μ_A are unknown, and the goal is to compute the

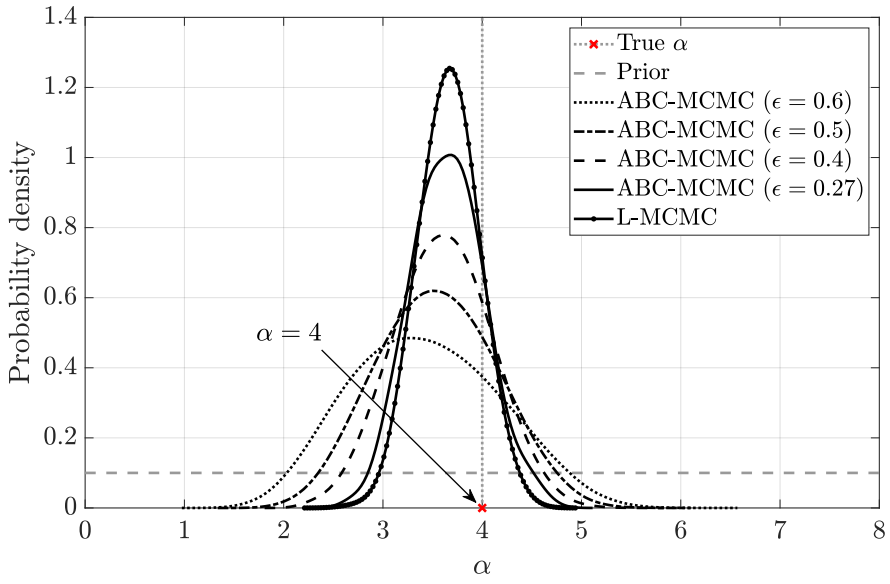


Figure D.4: Posterior distributions of α produced by ABC-MCMC and L-MCMC.

joint and marginal posteriors of these two parameters using the synthetic data set. The prior distributions are considered diffused uniform priors, i.e., $f(\eta, \mu_A) = f(\eta)f(\mu_A)$, where $f(\eta) = \mathcal{U}[0, 5]$ and $f(\mu_A) = \mathcal{U}[0, 1]$. For η and μ_A, the proposal distribution can be conveniently selected as independent normal distributions centered at the current samples with COVs equal to 0.01 and 0.1, respectively.

The ABC-MCMC is employed using $\epsilon = 0.61$ as the tolerance threshold. The initialization of the ABC-MCMC run using Algorithm 13 is conducted through the same sequence of threshold values as used in Simulation example 1. For the L-MCMC method, 5000 Monte Carlo samples are considered for likelihood evaluation. Both Bayesian inference schemes generated three chains. The chosen chain lengths, burn-in lengths and thinning intervals along with the computation times are presented in Table D.2. The computation times are given for single MCMC runs. These chain attributes are selected based on the MCMC traceplots, and their corresponding ACF and GR statistic plots as shown in Figure D.5 and Figure D.6. The 0.01th percentile

value estimated from the distribution of the proposed distance values is 0.622, which proves that the chosen threshold $\epsilon = 0.61 < 0.622$ satisfies the tolerance selection criterion.

Table D.2: Selected attributes of the MCMC chains and computation times.

Method	Tolerance (ϵ)	Chain length	Burn-in length	Thinning interval	Computation time
ABC-MCMC	0.61	2×10^7	–	50000	1790 s
L-MCMC	–	1×10^3	100	4	9125 s

Figure D.7 depicts the marginal posteriors and scatter plot of the joint posterior of η and μ_A obtained from both Bayesian computation schemes. The means, COVs and 95% CIs of the marginal posterior distributions are presented in Table D.3. The results produced by the likelihood-free method show great similarity to the results obtained from the likelihood-based method. But ABC-MCMC turns out to be extremely efficient in this problem with an overall 400% more efficiency than the likelihood-based method.

Table D.3: Summary statistics of the posterior parameter distributions.

Parameter	True value	ABC-MCMC			L-MCMC		
		Mean	COV	95% CI	Mean	COV	95% CI
η	2.5	2.5052	0.0101	[2.4540, 2.5513]	2.5085	0.0081	[2.4652, 2.5472]
μ_A	0.5	0.4448	0.1312	[0.3296, 0.5559]	0.4485	0.0995	[0.3625, 0.5332]

D.3 Example 3: The Six-Parameter Model

This example attempts to estimate all six parameters of the proposed gamma process model of degradation. The same synthetic data set used in Example 2 is considered once again. The chosen independent prior distributions are: $f(\alpha) = \mathcal{U}[0, 5]$, $f(\eta) = \mathcal{U}[0, 5]$, $f(\beta) = \mathcal{U}[0, 1]$, $f(\mu_A) = \mathcal{U}[0, 1]$, $f(\sigma_A) = \mathcal{U}[0, 1]$, and $f(\sigma_Z) = \mathcal{U}[0, 1]$. Independent normal distributions with means equal to current samples and COVs equal to 0.1 are chosen as proposal distributions. For ABC-MCMC, the initial proposal distributions

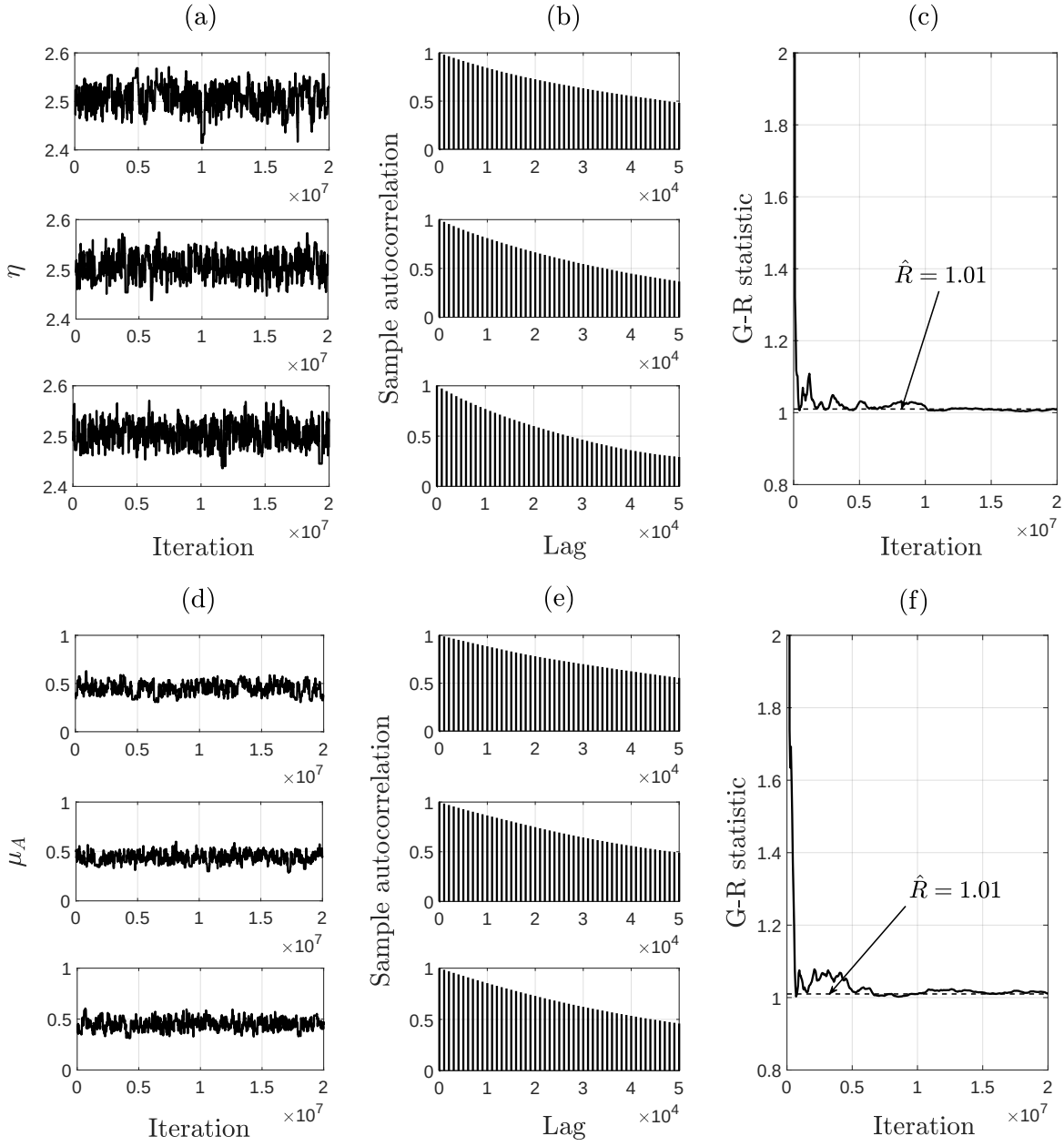


Figure D.5: Three MCMC runs of the parameters (a) η and (d) μ_A generated by ABC-MCMC with $\epsilon = 0.61$, and the corresponding (b,e) sample ACF plots, and (c,f) the convergence of the GR statistics.

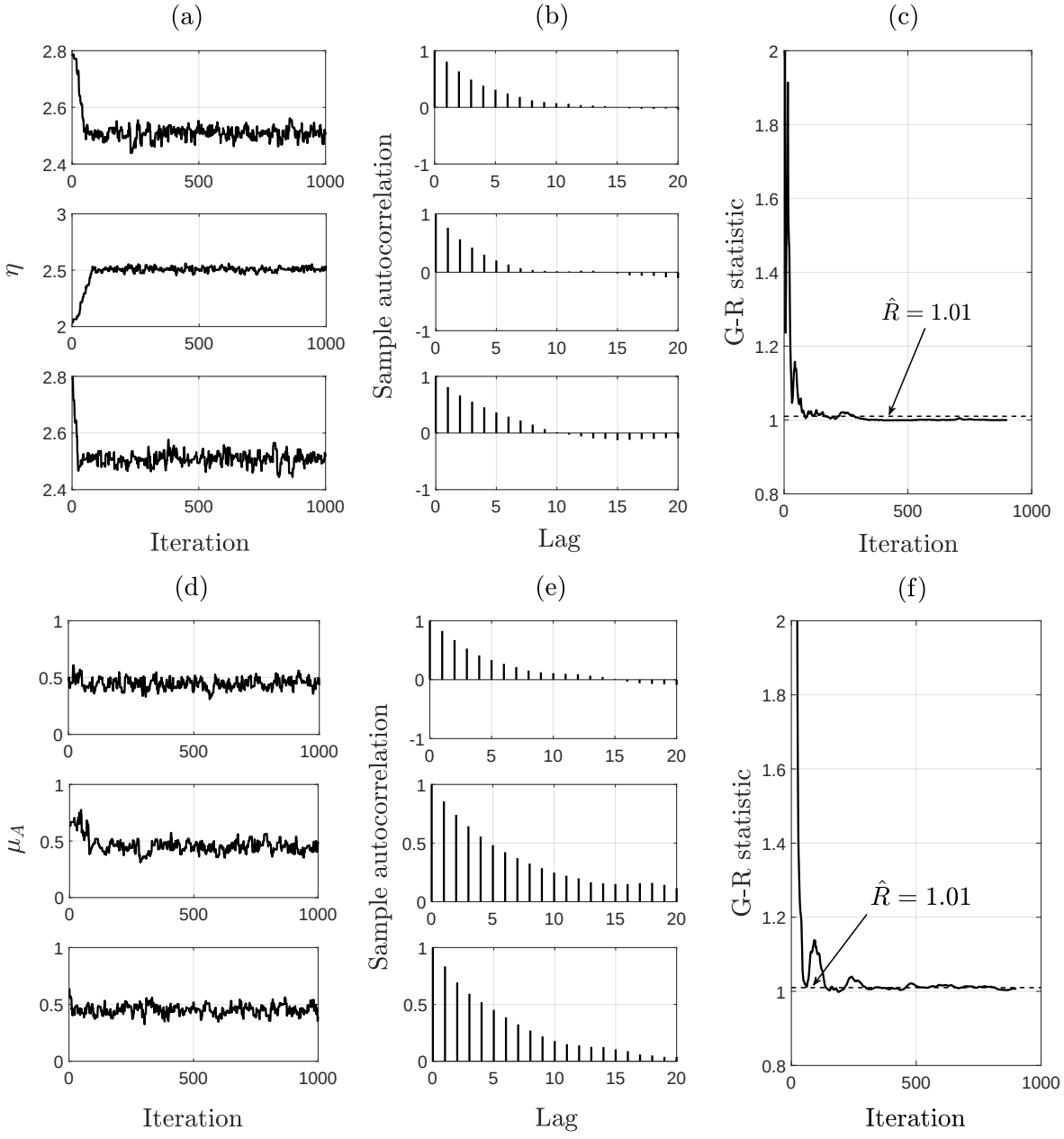


Figure D.6: Three MCMC runs of the parameters (a) η and (d) μ_A generated by L-MCMC, and the corresponding (b,e) sample ACF plots, and (c,f) the convergence of the GR statistics.

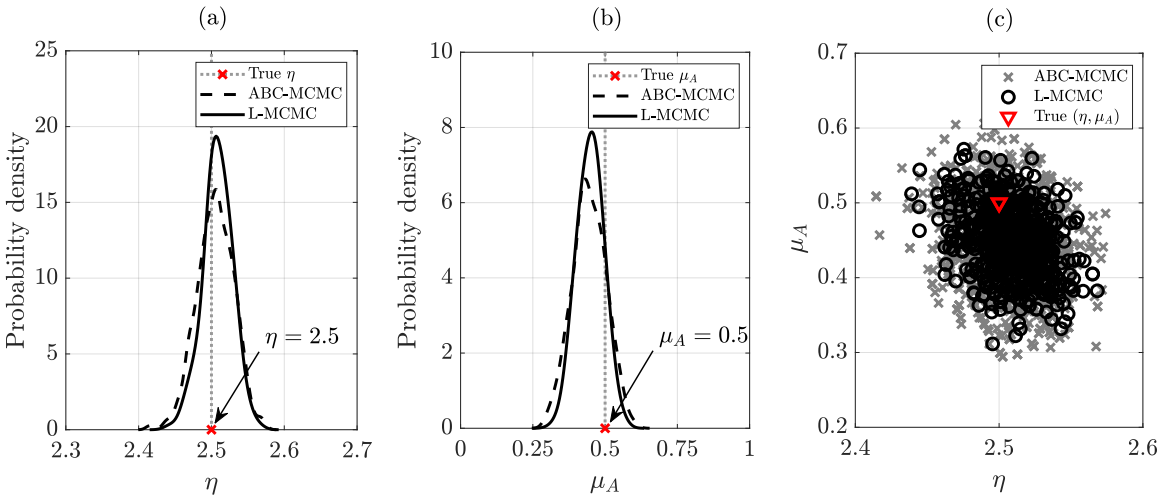


Figure D.7: Marginal posterior distributions of (a) η and (b) μ_A and (c) the joint posterior distributions as two-dimensional scatter plots produced by ABC-MCMC and L-MCMC.

are chosen to have twice the COVs of the proposals used in the main algorithm. A tolerance threshold of $\epsilon = 0.52$ is selected for ABC-MCMC. Due to the high dimension of the parameter space, the acceptance rate of both methods is found to be very low, thus resulting in very long MCMC chains. To remedy this, the model parameters are updated one-by-one sequentially; this reduces the rejection rate in both methods.

The likelihood function in L-MCMC is integrated using 5000 Monte Carlo samples. L-MCMC spent around 7 hours to generate a single chain of 3000 iterations length. A total of three MCMC chains are generated, and all of them converged at around 1700 iterations; hence a burn-in length of the same amount was chosen. Similarly, ABC-MCMC generated a single chain of 5×10^7 iterations length in around 2 hours and 40 minutes.

The marginal posterior distributions of the model parameters are shown in Figure D.8 and their summary statistics are presented in Table D.4. It can be observed that ABC-MCMC and L-MCMC produced comparable results. Marginal posteriors of α produced by both methods seem to be very close; however, a noticeable difference

in posterior locations can be spotted for parameters η, β, and μ_A. The reason behind this anomaly is the selected threshold value not being zero, resulting in an approximation of the joint posterior. Moreover, it can be noticed that the standard deviation parameters, σ_A and σ_Z, are difficult to estimate using both methods.

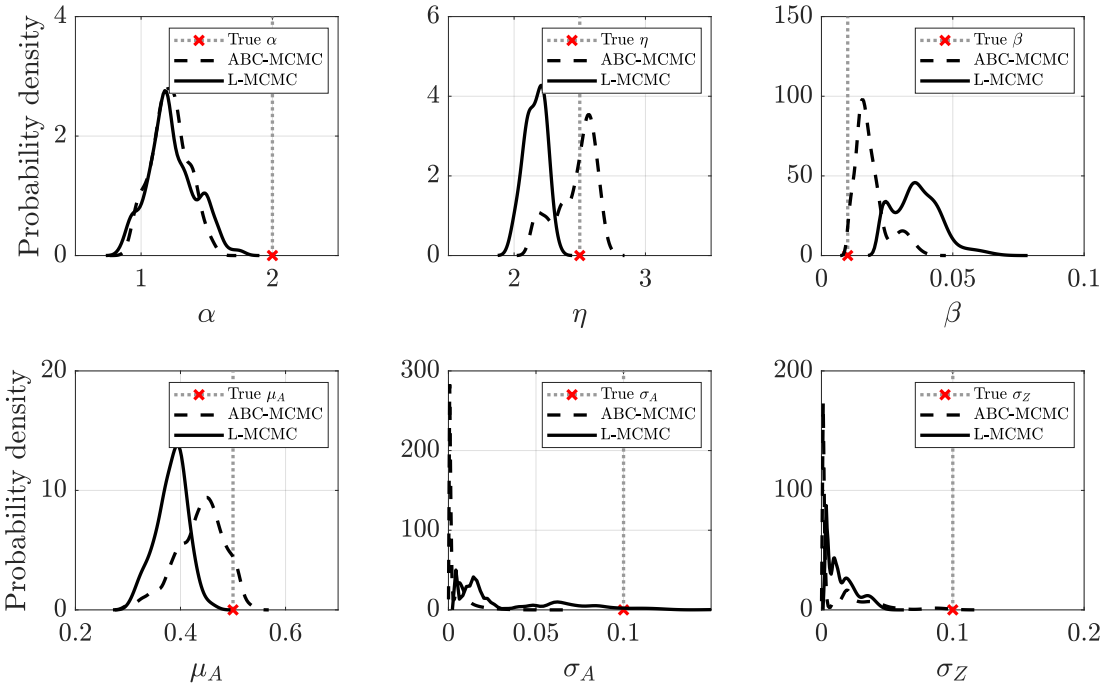


Figure D.8: Marginal posterior distributions of the model parameters produced by ABC-MCMC and L-MCMC.

Table D.4: Summary statistics of the posterior parameter distributions.

Parameter	True value	ABC-MCMC			L-MCMC		
		Mean	COV	95% CI	Mean	COV	95% CI
α	2	1.2171	0.1144	[0.9583, 1.4901]	1.2352	0.1462	[0.9172, 1.5842]
η	2.5	2.4693	0.0614	[2.1595, 2.6560]	2.1603	0.0357	[2.0046, 2.2922]
β	0.01	0.0187	0.3217	[0.0109, 0.0335]	0.0364	0.2401	[0.0227, 0.0566]
μ_A	0.5	0.4366	0.1056	[0.3268, 0.5118]	0.3831	0.0780	[0.3218, 0.4370]
σ_A	0.1	0.0032	1.6477	[0.0010, 0.0200]	0.0376	0.9049	[0.0035, 0.1168]
σ_Z	0.1	0.0139	1.4748	[0.0012, 0.0880]	0.0160	0.6996	[0.0025, 0.0403]

To estimate the lifetime distribution of the component population, a critical limit of 4 mm degradation is selected. With the help of simulation, the mean and fifth

percentile of population lifetime are estimated and presented in Figure D.9. The statistical properties of the lifetime distribution are shown in Table D.5. The true values of the lifetime quantiles are estimated using the true model parameter values. Both the likelihood-based and likelihood-free approaches produced almost similar distributions for the mean and fifth percentile of lifetime estimates. Moreover, both methods show excellent accuracy in comparison to the estimates obtained using the true values of the model parameters. Once again, ABC-MCMC proves to be highly efficient compared to L-MCMC.

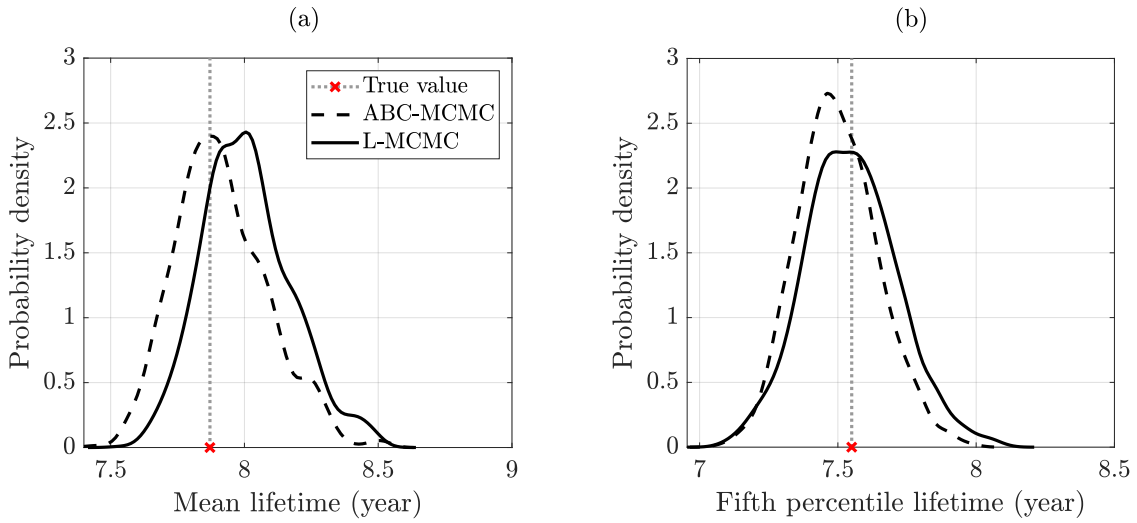


Figure D.9: (a) Mean and (b) fifth percentile of the lifetime distribution produced by ABC-MCMC and L-MCMC.

Table D.5: Statistical properties of the lifetime distribution.

Lifetime	True value (year)	ABC-MCMC			L-MCMC		
		Mean (year)	COV	95% CI	Mean (year)	COV	95% CI
Mean	7.87	7.91	0.0219	[7.62, 8.28]	8.01	0.0208	[7.71, 8.40]
Fifth percentile	7.55	7.49	0.0195	[7.25, 7.80]	7.55	0.0221	[7.23, 7.89]

Appendix E

Distribution of Maximum Pit Depth

The probability distribution of the maximum pit depth can be derived analytically as follows. Let us assume that the true number of pits generated between time 0 and t is $N(t) = n$. Given the number of pits at time t, assuming independence of the pit depths, the conditional CDF of the maximum pit depth can be written as

$$F_{\max}(h \mid n; t) = P[H_1 \leq h, H_2 \leq h, \cdots, H_n \leq h] = [F_H(h)]^n \tag{E.1}$$

where $H_1, H_2, \cdots, H_n$ are the depths of the n pits, and $F_H(h)$ is the CDF of the pit depth which can be written as

$$F_H(h) = 1 - \exp[-(h/\gamma)^\beta] \tag{E.2}$$

Since the number of pits $N(t)$ is a Poisson random variable and can vary between 0 and ∞, the CDF of the maximum pit depth can be written as

$$F_{\max}(h;t) = \sum_{n=0}^{\infty} F_{max}(h \mid n;t) P[N(t) = n] \tag{E.3}$$

Substituting Equation E.1 and the Poisson distribution from Equation 2.23 with a mean value of $\Lambda(t)$ into Equation E.3, we get

$$\begin{aligned} F_{\max}(h;t) &= \sum_{n=0}^{\infty} \frac{[\Lambda(t)F_H(h)]^n}{n!} \exp[-\Lambda(t)] \\ &= \exp[-\Lambda(t)\{1 - F_H(h)\}] \\ &= \exp[-\lambda t^{\delta} \exp\{-(h/\gamma)^{\beta}\}] \end{aligned} \tag{E.4}$$

Differentiating the CDF in Equation E.4 with respect to h, one can obtain the PDF of maximum pit depth as

$$\begin{aligned} f_{\max}(h;t) &= -\lambda t^{\delta}(-\beta/\gamma)(h/\gamma)^{\beta-1} \exp[-(h/\gamma)^{\beta}] \exp[-\lambda t^{\delta} \exp\{-(h/\gamma)^{\beta}\}] \\ &= \Lambda(t) f_H(h) \exp[-\Lambda(t)\{1 - F_H(h)\}] \end{aligned} \tag{E.5}$$

Equation E.5 shows that the PDF of maximum pit depth has four parameters $\{\lambda, \delta, \gamma, \beta\}$, and it is a function of time. The equation can be used for predicting the maximum pit depth distribution once the posterior samples of the model parameters are available. Since the maximum pit depth has a non-negative support, one can calculate its expected value using the expression $\int_0^{\infty}[1 - F_{\max}(h;t)]dh$. Equation E.4 can be used to calculate the 95th percentile of the maximum pit depth.