Quantifying and Reducing Input Modelling Error in Simulation

Lucy E. Morgan, B.Sc.(Hons.), M.Res

Lancaster University

Submitted for the degree of Doctor of Philosophy at Lancaster University.

December 2018

EPSRC
Engineering and Physical Sciences
Research Council

STOR-i
excellence with impact

Abstract

This thesis presents new methodology in the field of quantifying and reducing input modelling error in computer simulation. Input modelling error is the uncertainty in the output of a simulation that propagates from the errors in the input models used to drive it. When the input models are estimated from observations of the real-world system input modelling error will always arise as only a finite number of observations can ever be collected. Input modelling error can be broken down into two components: variance, known in the literature as input uncertainty; and bias. In this thesis new methodology is contributed for the quantification of both of these sources of error.

To date research into input modelling error has been focused on quantifying the input uncertainty (IU) variance. In this thesis current IU quantification techniques for simulation models with time homogeneous inputs are extended to simulation models with non-stationary input processes. Unlike the IU variance, the bias caused by input modelling has, until now, been virtually ignored. This thesis provides the first method for quantifying bias caused by input modelling. Also presented is a bias detection test for identifying, with controlled power, a bias due to input modelling of a size that would be concerning to a practitioner. The final contribution of this thesis is a spline-based arrival process model. By utilising a highly flexible spline representation, the error in the input model is reduced; it is believed that this will also reduce the input modelling error that passes to the simulation output. The methods described in this thesis are not available in the current literature and can be used in a wide range of simulation contexts for quantifying input modelling error and modelling input processes.

This thesis is dedicated to Jack.

"When life gives you lemons. Don't make lemonade. Make life take the lemons back!"

- C. Johnson. *Portal 2*.

Acknowledgements

I would firstly like to thank STOR-i CDT for the numerous opportunities they have given to me throughout my MRes and PhD, I also gratefully acknowledge the financial support of the EPSRC. Being part of STOR-i has been an absolute pleasure. The support of the staff and students has helped me both academically and personally in many ways over the last four years. I'm grateful to all the people I've been lucky enough to share this process with. Doing a PhD is no easy task and I've loved being able to smile, laugh and, at times, cry with my STOR-i family. I'd like to add a special thanks to a few people in STOR-i who have made my time here brighter. To Kathryn for all the laughter, you've been a true friend. To Luke, my "academic brother", I wish you all the success in the years ahead. To Emma and Emily thank you for introducing me to the country music that helped me dance my way through the last few months. To Jake thank you for all the silliness, I know you're going on to great things. And finally to Helen and Katie, I can't be thankful enough I had two such great friends to pick me up when times were hard and to celebrate the best times.

Of course I wouldn't have left square one without the support of my three amazing supervisors. I've been so very lucky to work with you all. I'd firstly like to thank Dr Andrew Titman for all his help, patience and his deep understanding of statistics, I can't promise I won't be knocking on your door in future. Secondly I'd like to thank Dr David Worthington for his continuing support. I chose my PhD project because I felt we'd work well together, I wasn't wrong, and I hope there are opportunities to continue to work together in future. Lastly I'd like to thank Professor Barry Nelson for stretching my research ability and helping me to achieve higher than I ever thought possible. Taking on a student from a different

continent was a large commitment, I hope you've enjoyed the process as much as I have! Special thanks also go to Jeanne Nelson for giving me such a warm welcome on my visits to Chicago.

The PhD process has seen me go through some of my biggest ups and downs and so a massive thank you has to go to my partner in crime Alec for making me smile daily. I wouldn't be at the finish line without your support. Seeing you finish your PhD I knew this process would be hard and I'm so glad I had you by my side through it all. I'm excited to see what our future holds.

My final thanks go to my wonderful family! To my best friend and big brother Joshua I have a lot to thank. I don't know where either of us would be if we weren't so competitive. Thank you for always inspiring me to keep reaching higher and to challenge myself for better or worse. And to my Mum and Dad I give the biggest thank you of all. I owe you so much for the way you brought me up and the support and encouragement you've given me to achieve my goals. Without your love, guidance and time I wouldn't be where I am and I appreciate every bit of it.

Declaration

I declare that the work in this thesis has been done by myself and has not been submitted elsewhere for the award of any other degree.

This thesis is constructed as a series of papers. Chapters 3, 4 and 5 should therefore be read as separate entities.

Chapter 3 has been published as Morgan, L. E., Nelson B. L., Titman A. C. and Worthington D. J. (2016). Input Uncertainty Quantification for Simulation Models with Piecewise-constant Non-stationary Poisson Arrival Processes. In *Proceedings of the 2016 Winter Simulation Conference*, pages 370-381. IEEE Press.

An early version of Chapter 4 was published as Morgan, L. E., Nelson B. L., Titman A. C. and Worthington D. J. (2017). Detecting Bias due to Input Modelling in Computer Simulation. In *Proceedings of the 2017 Winter Simulation Conference*, pages 1974-1985. IEEE Press. An extended version of this paper was then submitted for publication as Morgan, L. E., Nelson B. L., Titman A. C. and Worthington D. J. (2018). Detecting Bias due to Input Modelling in Computer Simulation. *European Journal of Operational Research*. This submission is currently under revision.

The word count for this thesis is 42,131 words.

Lucy E. Morgan

Contents

List of Figures

List of Tables

Introduction

1.1 Motivation

Stochastic simulation is a tool used to aid decision making. It allows practitioners to analyse and experiment with systems that are driven by random processes. For systems where the performance measures of interest are mathematically intractable, stochastic simulation is a natural choice. In practice simulation is used in many industries to study complex systems, examples include: healthcare (Brailsford, 2007), aviation modelling and analysis (Rhodes-Leader et al., 2018) and manufacturing (Law, 1988).

The conclusions drawn from simulation experiments are conditional on the input models that drive them. Typically these input models, represented by probability distributions or processes, are estimated using observations collected from the real-world system using statistical methods such as maximum likelihood estimation. When this is the case uncertainty arises in the estimated input models due to the fact that only a finite number of observations can be collected from the system of interest. As the amount of input data increases the error in the input models decreases, but they are never perfectly correct. In experiments where constraints on time and money have limited the number of observations collected from a system, the error in the input models can be substantial. For example, in a manufacturing context the pressure to make a timely decision about whether to switch to a new method of production may limit the time available to observe and model the various manufacturing processes.

In this thesis the error that propagates through a simulation model, from the estimated input models to the performance measures under study, is referred to as the error caused by input modelling. In practice ignoring error caused by input modelling can lead to over-confidence in the decisions supported by the simulation. The problem of quantifying and reducing the error in the output of a simulation caused by error in the input models therefore motivates this thesis.

In recent years there has been substantial interest in quantifying the variance caused by input modelling in a simulation response. In the simulation community this variance is known as input uncertainty. Unlike stochastic uncertainty, which can be reduced by performing additional replications of the simulation, input uncertainty can only be reduced by collecting further observations of the system to gain better estimates of the input models. Methods for input uncertainty quantification for simulation models with homogeneous inputs exist, and one such method, proposed by Song and Nelson (2015), has been implemented in the commercial software Simio (2015). Despite nonhomogeneous input models commonly being used in simulation experiments, input uncertainty quantification for non-homogeneous input models is yet to be addressed. One focus of this thesis is therefore the quantification of input uncertainty for simulation models with non-homogeneous Poisson processes.

Interest in error caused by input modelling has, until now, been focused on input uncertainty quantification, but estimating the input models that drive the simulation also causes bias in the simulation response. Bias caused by input modelling arises when the simulation response is a non-linear function of its inputs, which is usually the case in the complex systems for which simulation is used. As the number of observations available to estimate an input model tends to infinity, bias caused by input modelling is known to decrease faster than input uncertainty. This knowledge has previously been used to justify ignoring bias caused by input modelling, but when the number of observations is finite nothing can be said about the relative size of bias and input uncertainty and it therefore should not be ignored.

In simulation in practice a common way of representing a nonhomogeneous arrival process is to use a nonhomogeneous Poisson process (NHPP). Substantial research has been carried out into fitting the arrival rate function, $\lambda(t)$, and integrated rate function, $\Lambda(t)$, of a NHPP to observed data, and for simulating arrivals from these representations using techniques such as thinning and inversion. The common use of NHPPs in practice has also led to their availability in commercial simulation software. Given the value of NHPPs as input models to simulation there is a motivation to create an input modelling method that recovers the true arrival rate function of a NHPP "better" than existing methods. By reducing the error between the true input model and the fitted model a reduction in the error caused by input modelling should be seen in the simulation output. Providing practitioners with the tools to quantify and reduce the error in the simulation output caused by input modelling would improve their ability to make decisions with the support of simulation.

The outputs of this thesis are threefold. First, new methodology for the quantification of input uncertainty in simulation models with non-stationary input processes is presented. Second, new methodology for detecting the bias caused by input modelling on the output of simulation is presented. Finally, a new spline-based input modelling method for the arrival rate of a NHPP is developed.

1.2 Contributions

We now outline for the reader the main contributions of this thesis.

We first contribute two methods for quantifying input uncertainty for simulation models with nonhomogeneous inputs. These methods extend the techniques of Cheng and Holland (1997) and Song and Nelson (2013) for quantifying input uncertainty in systems with time homogeneous inputs. Specifically we focus on simulation models with piecewise-constant non-stationary Poisson arrival processes. In practice arrival processes to simulation models are often nonhomogeneous with respect to time. Numerical evaluation and illustrations of the methods are provided and indicate that the methods perform well.

Our second contribution is to provide the first method for quantifying bias caused by input modelling. This also provides the first way to summarise the mean squared error caused by input modelling for a simulation performance measure by bringing together the input uncertainty variance and the squared bias. As the key to this contribution a bias detection test is also presented with controlled power for detecting bias of a size that exceeds a threshold deemed to be concerning by a practitioner. We numerically evaluate the bias detection test and demonstrate its use in a realistic case study concerning a healthcare call centre.

The final contribution of this thesis is a spline-based arrival process modelling method. Specifically we develop a new method for representing the arrival rate function of a NHPP and a simple method for simulating arrivals from it. By using a spline function representation with a large number of knots we reduce the bias, with respect to the true arrival rate, in the model. The more knots used to build the spline function the more flexible it can become, we therefore control over fitting, and thus variability, by penalising the NHPP log-likelihood when fitting the spline function. By aiming to reduce the error in the arrival process model, we also reduce the input modelling error passed to the simulation output. To evaluate this model we compare it to the methods of Zheng and Glynn (2017) and Chen and Schmeiser (2017), from the arrival process modelling literature, and demonstrate the use of the spline-based input model using observations from a real-world A&E department with a cyclic arrival rate function.

1.3 Outline of Thesis

The thesis is now outlined for the reader. In Chapter 2 the key concepts, terminology and methodology required within this thesis are introduced. In Chapter 3 we present two new methods for input uncertainty quantification in simulation models with piecewise-constant nonhomogeneous Poisson arrival processes. In Chapter 4 we approach the issue of detecting bias due to input modelling in stochastic computer simulation. In Chapter 5 we present a spline-based input model of the arrival rate function of a NHPP and a simple method for simulating arrivals from it. Finally, in Chapter 6 the thesis is concluded with a summary of contributions and some ideas for further work in the area of quantifying and reducing error caused by input modelling.

Background Material

In this chapter the reader is provided with the necessary background material, including references to useful sources, to aid the understanding of this thesis. Firstly the concept of nonhomogeneous Poisson processes (NHPPs), how to check real-world data follows a NHPP and techniques for simulating data from a NHPP are introduced. Secondly, input modelling is discussed with specific detail on modelling the arrival rate and integrated rate functions of a NHPP. The idea of error caused by input modelling is then introduced alongside definitions of input uncertainty and bias caused by input modelling. Finally spline functions and B-spline basis functions are introduced.

2.1 Nonhomogeneous Poisson processes

The focus of this thesis is on the simulation of systems with input processes that can be appropriately described by non-homogeneous Poisson processes (NHPPs). In reality, arrivals to a system are often known to be non-stationary; NHPPs are a common model used to describe the arrival process when this is the case. For example, Pritsker et al. (1996) used NHPPs for fitting donor and patient arrivals within a large scale simulation model developed for the United Network of Organ Sharing (UNOS). NHPPs can be used to model many types of non-stationary arrival process and are therefore appropriate for use in many application areas including: manufacturing (Viswanadham and Narahari, 1992), healthcare (Green, 2006), and call centres (Kim and Whitt, 2014).

A NHPP is a generalisation of a homogeneous Poisson process. For a homogeneous Poisson process events are said to occur at a constant rate λ per unit time. In a NHPP this rate, or intensity, $\lambda(t)$, is allowed to change through time and is assumed non-negative, $\lambda(t) \geq 0$, for all t (Kingman, 1992). Given two points a and b, where $a \leq b$, let N be a NHPP and $N(a,b)$ denote the number of events on interval $(a,b]$. By definition of a NHPP, the number of observations in interval $(a,b]$, $N(a,b)$, follows a Poisson distribution

$$N(a,b) \sim \mathrm{Pois}(\Lambda(a,b))$$

where the probability of s events occurring on interval $(a,b]$ is

$$\mathrm{P}(N(a,b) = s) = \frac{\exp\{-\Lambda(a,b)\}\Lambda(a,b)^s}{s!}. \tag{2.1.1}$$

Here $\Lambda(a,b)$ is known as the integrated rate, or cumulative intensity function and is defined by

$$\Lambda(a,b) = \int_a^b \lambda(t)dt.$$

Within interval $(a,b]$, $\Lambda(a,b)$ can be interpreted as the expected number of observations $\Lambda(a,b) = \mathrm{E}[N(a,b)]$. When considering the expected number of observations up to time t, $\Lambda(0,t) = \int_0^t \lambda(s)ds$ let the integrated rate function be denoted by $\Lambda(t)$.

Another property of a NHPP is that the sum of q independent NHPPs is also a NHPP,

$$N = N_1 + N_2 + \cdots + N_q,$$

where N_i for $i = 1,2,\ldots,q$ are NHPPs, see Blumenfeld (2009). When this is the case the rate, or intensity, function of N can also be decomposed into the sum of the intensity functions of its q components

$$\lambda^c(t) = \lambda_1^c(t) + \lambda_2^c(t) + \cdots + \lambda_q^c(t),$$

where $\lambda_i^c(t)$ is the rate function from NHPP N_i for $i = 1,2,\ldots,q$. This decomposition is used in Chapter 5 to aid generation of arrivals from a spline function.

In simulation it is important to model an input process as nonhomogeneous if it is so. Arrivals to systems in the real-world are often seen to vary through time. Estimating the distribution of these arrivals using a homogeneous distribution would remove any fluctuations in the arrival process; this may have a large impact on the output measures of the simulation. For example, in a call centre there are usually times of the day where the arrival rate of calls peaks and troughs. A constant arrival rate would therefore not represent the true arrival rate to this system well; and would lead to over or under staffing, see Whitt (2007). In Chapter 3 the arrival rate to a healthcare call centre is used to guide the number of staff to have on duty. This can be seen to have a large effect on the expected waiting time of callers when comparing the use of homogeneous and nonhomogeneous arrival processes as inputs to the simulation.

2.1.1 Real-world observations

For a NHPP, denoted N, the number of arrivals during interval $(a,b]$, denoted $N(a,b)$, follows a Poisson distribution. Therefore the dispersion, or variance-to-mean ratio, of the number of arrivals during an interval, $\omega = \mathrm{E}\left[N(a,b)\right]/\mathrm{Var}\left[N(a,b)\right]$, should equal 1. This is not guaranteed given real-world data, even when the underlying process is a NHPP. Arrival processes in practice can be both under ($\omega < 1$) or over ($\omega > 1$) dispersed in comparison to a NHPP. It is therefore sensible to perform checks on the real-world observations to confirm the appropriateness of modelling an input as an NHPP.

One way to check whether the observed data within an interval, $(a,b]$, follow a NHPP is to record the arrival counts to the interval multiple times and use a chi-square goodness-of-fit test to check whether this data comes from a Poisson distribution. The chi-square test for supposedly Poisson data compares the observed counts over an interval, O_i, to the expected count, E_i, assuming the data came from a Poisson distribution. For example, given observations of arrivals to an A&E department on Mondays over w weeks the following hypothesis

$$\mathrm{H}_0: \text{The total number of arrivals on Monday is Poisson}$$

would be tested against

$$\text{H}_1 : \text{The total number of arrivals on Monday is not Poisson.}$$

with test statistic, T,

$$T = \sum_{i=1}^{w} \frac{(O_i - E_i)^2}{E_i}.$$

Note that the expected number of arrivals on a Monday can be estimated by the mean of the observed counts $E_i = \bar{O} = \frac{1}{w}\sum_{i=1}^{w} O_i$. In comparing the test statistic, T, to the critical value ψ, of the chi-squared distribution with $w-1$ degrees of freedom at the $\alpha\%$ significance level, if $T < \psi$ there is not significant evidence to reject the null hypothesis that the observed counts are from a Poisson distribution. Of course the chi-square goodness-of-fit test is not a guarantee that the observed data is Poisson but, when the null hypothesis is rejected, this indicates that the data is not Poisson; the test is therefore a good warning tool. Another consideration in using the Chi-squared test is that the number and location of the intervals may not be known. In practice they may have to be chosen by the practitioner which introduces subjectivity into the approach.

2.1.2 Generating arrivals

Simulating a homogeneous Poisson process is relatively simple. The inter-arrival time between consecutive customers is known to be an exponential random variable with cumulative density function (cdf) $F(t) = 1 - \exp\{-\lambda t\}$, $t \geq 0$. Simulation of the i^{th} customers arrival time is therefore simply

$$y_i = y_{i-1} + F^{-1}(s) = y_{i-1} - \frac{\ln(1-u_i)}{\lambda}$$

where u_i is a random variable generated from the Uniform(0,1) distribution, $u_i \sim \text{Uniform}(0,1)$.

Simulation of arrivals from a NHPP is not as easy due to the varying arrival rate function, $\lambda(t)$. Two methods for simulating arrivals from NHPPs are inversion and thinning.

Inversion

Inversion is so called due to its dependence on the the inverse of the integrated rate function, $\Lambda^{-1}(t)$. Cinlar (2013) proves that random variables T_i, $i = 1, 2, \dots$ are arrival times from a NHPP N with integrated rate function $\Lambda(t)$ if and only if $\Lambda(T_1), \Lambda(T_2), \dots$ are event times from a stationary Poisson process with rate one. This holds for all $t \geq 0$ when $\Lambda(t)$ is a positive-valued, continuous, non-decreasing function. Inversion therefore requires the integrated rate function $\Lambda(t)$ to be invertible. When it is possible to calculate this inverse, the method proceeds as follows:

1. Generate random variable $u_i \sim \text{Uniform}(0, 1)$.
2. Generate Poisson arrival times with rate $\lambda = 1$ using $y_0 = 0$, $y_i = y_{i-1} - \ln(1 - u_i)$.
3. Calculate the i^{th} arrival time from the NHPP, t_i, by $t_i = \Lambda^{-1}(y_i)$.

One problem with inversion is that the integrated rate function, $\Lambda(t)$, will not always have a tractable inverse. Although there may be no tractable functional form for $\Lambda^{-1}(t)$, the integrated rate function will always be numerically invertible. Numerical inversion in this case is equivalent to a one-dimensional search. To account for possible flat regions of the integrated rate function a generalised definition of the inverse integrated rate function is used,

$$\Lambda^{-1}(t) = \inf\{x \in \mathbb{R} : \Lambda(x) \geq t\}.$$

For generating a single arrival the numerical search for the inverse should be reasonably quick but, within a simulation model, it is quite possible that thousands of arrivals will be required in which case completing a search for each arrival will add up. A common approach to improve efficiency in this case is to numerically evaluate $\Lambda^{-1}(t)$ over a grid of points and linearly interpolate between these.

When $\Lambda(t)$ is easily invertible, inversion is a simple efficient way of generating arrivals from a NHPP. An example of this is presented by Klein and Roberts (1984) who show that for a NHPP with a piecewise-linear rate function, $\lambda(t)$, the integrated rate function, $\Lambda(t)$, is piecewise-quadratic within each interval and thus a tractable inverse function exists from which to generate arrivals from the underlying NHPP.

Thinning

Thinning is an arrival generation method for NHPPs that works directly with the rate function, $\lambda(t)$. The key idea is to generate arrivals from an alternative function that is both simpler to generate arrivals from and that majorises the original function of interest $\lambda(t)$. The arrivals generated from the alternative function are then 'thinned', some are thrown out, according to the probability that they came from the NHPP with arrival rate $\lambda(t)$.

Like inversion, thinning traditionally begins with the generation of arrivals from a stationary Poisson process but in this case the stationary arrival process has arrival rate equal to the maximum rate, $\max_t \lambda(t) = \lambda^\star$. Arrivals are discarded according to the probability of them having come from the NHPP. Specifically an arrival at time t is rejected according to a Bernoulli trial with success probability $\lambda(t)/\lambda^\star$. The probability that a potential arrival is thinned is thus $1 - \lambda(t)/\lambda^\star$. The discrepancy between $\lambda^\star$ and the arrival rate function of interest $\lambda(t)$ has a large effect on how many arrivals are rejected and thus the efficiency of the method. For an algorithm of how to implement thinning see Nelson (2013). A proof that the thinning method samples from a NHPP with rate $\lambda(t)$ is omitted here but a sketch proof can be found in Kuhl and Wilson (2009). One advantage of this approach is that thinning can be used to generate arrivals from any bounded arrival rate function $\lambda(t)$; complexity is not an issue. Although, it may be said that thinning is a wasteful method. For example, if the rate function of interest has a high peak with short duration and the rest of the process is a much lower rate then thinning using $\lambda^\star = \max_t \lambda(t)$ could be highly inefficient, discarding a high proportion of simulated points.

By using the inversion method of Klein and Roberts (1984) to generate arrivals efficiently from a piecewise-linear rate function thinning can, in some cases, be made much more efficient. A piecewise-linear function, in most cases, can create a much tighter majorising function than a constant function. Also arrivals generated from a piecewise-linear majorising function can be thinned in the same way as the arrivals from a constant majorising function. Let $\tilde{\lambda}(t)$ denote the piecewise-linear majorising function, then the probability of thinning a potential arrival is $1 - \lambda(t)/\tilde{\lambda}(t)$. As before, thinning arrivals generated from $\tilde{\lambda}(t)$ gives arrivals from the arrival process with arrival rate $\lambda(t)$.

The methods of thinning and inversion both lead to an arrival process with the specified arrival rate $\lambda(t)$, but this does not mean that the arrivals generated using the methods will be the same. This is due to the stochastic variability in how the arrivals are generated between the two methods. Methods for modelling the rate, $\lambda(t)$, or integrated rate, $\Lambda(t)$, function of a NHPP will now be discussed.

2.2 Input Modelling

In this thesis "input modelling" refers to the method of forming a representation of an input to a simulation model from which event times can be generated. The focus of this thesis is on the estimation of input models using data. Sometimes data are unavailable and subjective decisions have to be made about certain inputs; from here on in inputs created in this way are not considered and focus lies on input models that have been estimated using observations from the system of interest.

This section reviews relevant methods within the input modelling literature with specific focus on methods for modelling and generation of NHPPs, as introduced in §2.1. For a more general discussion of input modelling techniques for use in discrete event simulation see Leemis (2001) and Cheng (2017), see also Cheng (1994) for a discussion of how to select appropriate input distributions. When arrival observations are over or under dispersed compared to a Poisson process, Gerhardt and Nelson (2009) present methodology for modelling non-stationary non-Poisson arrival processes and Nelson and Gerhardt (2011) consider the modelling and simulation of non-stationary, non-renewal processes. For a discussion of input modelling for complex problems see Nelson and Yamnitsky (1998).

Since the probabilistic behaviour of a NHPP can be completely characterised by its rate function, $\lambda(t)$, or integrated rate function, $\Lambda(t)$ (Kuhl and Wilson, 2009), any input modelling approach for a NHPP aims to estimate one of these functions. The existing input modelling literature will now be summarised.

2.2.1 Estimating the rate function, $\lambda(t)$

A common, early, approach to estimating the intensity function, $\lambda(t)$, of a NHPP was to use an exponential form. Exponentiating the rate function ensures it is always non-negative, $\lambda(t) \geq 0$. This idea was first considered by Cox and A. W. Lewis (1966) who stated that a continuous rate function for a NHPP can be estimated arbitrarily closely with an exponential polynomial function. This idea was built upon by Lewis (1971), Lewis and Shedler (1976), Lee et al. (1991) and Kuhl et al. (1997). The key idea is to model the intensity function by fitting an exponential function with some additional components to reflect knowledge about the underlying process. For example, Kuhl et al. (1997) present the exponential-polynomial-trigonometric rate function with multiple periodicities (EPTMP)

$$\lambda(t) = \exp\left(\sum_{j=0}^{l} \alpha_j t^j + \sum_{k=1}^{p} \gamma_k \sin(\omega_k t + \phi_k)\right), \tag{2.2.1}$$

which can handle NHPPs where the arrival rate exhibits trends and multiple-periodicities. All exponential forms of the rate function are parametric and require the estimation of parameters when being fit to data. For example, for the EPTMP rate function (2.2.1), estimation of the parameters $\{\alpha_0, \alpha_1, \ldots, \alpha_l, \gamma_1, \gamma_2, \ldots, \gamma_p, \omega_1, \omega_2, \ldots, \omega_p, \phi_1, \phi_2, \ldots, \phi_p\}$ would be required to fit the rate function. Numerically optimising these parameters is computationally expensive and often requires a good starting point.

Another common approach to modelling both the rate function, $\lambda(t)$, and the integrated rate function, $\Lambda(t)$, is to assume they take a piecewise form. Henderson (2003) considered piecewise-constant estimators of the rate function and showed that, when the intervals are chosen to be of equal length, this estimator is consistent as the number of intervals increases provided the length of the intervals shrink at an appropriate rate. In practice piecewise-constant estimators are popular, amongst the many examples of their use are Brown et al.

(2005) and Avramidis et al. (2004) who both use a piecewise-constant arrival rate in a call centre setting.

Massey et al. (1996) present a piecewise-linear representation of the arrival rate function given arrival count data. They fit the rate function using ordinary least squares (OLS), iterative weighted least squares (IWLS) and maximum likelihood (ML) where all methods are constrained to yield a non-negative rate function. More recently, Nicol and Leemis (2014) used count observations to provide a piecewise-linear estimator of the rate function, $\lambda(t)$. This method was formulated as a constrained quadratic programming problem with constraints on the continuity of the estimator, the estimator's mean value within an interval and optional constraints on the interval end points for cyclic contexts.

Chen and Schmeiser (2013) present an iterative algorithm for smoothing a piecewise-constant representation of the intensity function. Within each iteration they run their algorithm, Smoothing via Mean-constrained Optimized-Objective Time Halving (SMOOTH), which takes a piecewise constant arrival rate function and yields a 'smoother' representation with double the number of intervals, each with half the length. Here smoothness is measured in terms of the integrated squared second derivatives. The resulting representation is non-negative and maintains the integral of the original function and thus the expected number of arrivals in each interval. Iteration of the SMOOTH algorithm gives the proposed, I-SMOOTH method which returns a sequence of sequentially smoother arrival rate functions. The method is aimed to be automatic, but the user is required to set how many iterations to let I-SMOOTH carry out.

Note that the I-SMOOTH method requires the user to have a well chosen initial piecewise-constant representation of the arrival rate function. Chen and Schmeiser (2018) present a simple method for creating a piecewise-constant arrival rate function with an optimal number of equal length intervals given arrival observations by minimising an unbiased estimator of the mean integrated squared error (MISE). They show that the optimal number of intervals ι given a total of a arrival times is

$$\widehat{\iota} \equiv \text{argmin}_{\iota=1,2,\ldots}\, \iota(2a - \sum_{j=1}^{\iota}(C_j(\iota)^2))$$

where $C_j(\iota)$ is the number of arrivals in the j^{th} interval when there are ι intervals. This method could act as a pre-processing step for input modelling modelling methods, like I-SMOOTH, that assume the underlying NHPP has a piecewise form and that the number of intervals and interval locations are known.

As for higher order polynomial representations. Kao and Chang (1988) present a piecewise polynomial representation by ‘grafting’ polynomials of different degrees together whilst constraining the continuity of the resulting representation. The method is subjective in the choice of polynomial degree in each interval, and the break points at which the function changes.

In Chapter 5 a spline-based method for modelling and generating NHPPs is developed and compared to two recent methods in the literature: a piecewise-linear representation by Zheng and Glynn (2017) and a piecewise-quadratic representation by Chen and Schmeiser (2017). Both competing methods allow estimation of a NHPP intensity function from arrival time observations and both can be used alongside the pre-processing method of Chen and Schmeiser (2018).

Zheng and Glynn (2017) assume the underlying rate function of the NHPP is piecewise-linear and that the placement and number of intervals is known. They provide two methods for fitting the intensity function, one using maximum likelihood estimation, and one using ordinary least squares (OLS) methods. Given the assumed piecewise-linear form of the rate function they reduce the problem of estimating $\lambda(t)$ to estimating the arrival rate at the interval boundaries. This estimation is formulated as a convex, highly tractable optimisation problem which, provided there is at least one arrival in each interval, can be shown to have a unique solution. Both maximum likelihood and OLS methods are presented for observations in the form of arrival times and interval count data, and both formulations can handle cyclic rate functions. In the cyclic case for a rate function with p intervals the maximum likelihood optimisation problem is given by

$$\begin{aligned} \max_{y_0,y_1,\ldots,y_p} \quad & L_m(y_0,y_1,\ldots,y_p) \\ \text{s.t.} \quad & y_0 = y_p \\ & y_i \geq 0,\ 0 \leq i \leq p \end{aligned}$$

where $y_0, y_1, \ldots, y_p$ are the arrival rates at the interval boundaries and $L_m(\cdot)$ is the likelihood given m observations of the NHPP. The structure of this problem means that even for large m and moderate p the problem is computationally tractable.

Chen and Schmeiser (2017) present the Max Nonnegativity Ordering Piecewise-Quadratic Rate Smoothing (MNO-PQRS) algorithm, for general input processes, which takes a piecewise-constant representation of the rate function and returns a smoother, piecewise-quadratic function. The algorithm is not specific to NHPPs and requires no user specified parameters. It smooths the arrival rate function whilst maintaining the expected number of arrivals in each interval. MNO-PQRS has two components: first, the Piecewise-Quadratic Rate Smoothing (PQRS) algorithm smooths the initial piecewise-constant representation returning a continuous, differentiable rate function. Then, if PQRS returns any negative sections of the rate function, the Max Nonnegativity Algorithm (MNO) returns the maximum of zero and the PQRS representation. Like the I-SMOOTH method by Chen and Schmeiser (2013) the method requires an initial piecewise-constant representation to be known; this could be provided by the pre-processing method of Chen and Schmeiser (2018).

Another approach for fitting a NHPP rate function was presented by Kuhl and Bhairgond (2000) who construct a highly flexible NHPP rate function representation using wavelets. Their method has the advantage of requiring no prior knowledge or assumptions to be made about the behaviour of the process.

Channouf (2008) use a smoothing spline approach to represent the arrival rate of both a NHPP and a doubly stochastic Poisson process. The B-spline composition of a spline function is introduced in §2.4; Channouf (2008) use an alternative spline function composition where each component of the spline is represented by a polynomial of degree d. In fitting their spline function they use a recurrence relation to reduce their problem to a linear system of equations which they solve by Gaussian elimination.

2.2.2 Estimating the integrated rate function, $\Lambda(t)$

An alternative to estimating the rate function, $\lambda(t)$, of a NHPP is estimation of the cumulative rate function, $\Lambda(t)$. For the cumulative rate function there is a natural estimator, calculated from the observed data, known as the empirical cumulative rate function (ECRF). There is no counterpart of the ECRF for the rate function. The ECRF is a step function with each step corresponding to either an arrival to the system or a count of arrivals in an interval given observations of the input process. This method of modelling the cumulative rate function may be thought of as crude, but as the number of observations increases the bias reduces. The ECRF also has no dependency on the input process following Poisson assumptions.

When the input process is a NHPP, Leemis (1991) presents a non-parametric piecewise-linear estimator of the cumulative intensity function, $\Lambda(t)$. The method essentially linearly interpolates the ECRF function between ordered event times. Generation of event times from the NHPP given the piecewise-linear representation using inversion is also discussed. Arkin and Leemis (2000) extend this method to include overlapping realisations, the key to their method being to partition the observation interval into the smallest number of regions so that, within each region, there are a constant number of realisations observed.

Given observations of event counts Leemis (2004) presents a maximum likelihood estimator of the cumulative intensity function,

$$\widehat{\Lambda}(t) = \left(\sum_{j=1}^{i-1} \frac{n_j}{k} \right) + \frac{n_i\,(t-a_i)}{k\,(a_i-a_{i-1})} \qquad \text{for } a_{i-1} < t \leq a_i.$$

where $(a_0,a_1],(a_1,a_2],\ldots,(a_{f-1},a_f]$ are the f subintervals that partition the interval of observation and n_i denotes the number of observations over k realisations in interval i. In this method caution should be taken in the choice of the subinterval placement. If the subinterval lengths are too small, very few observations will be seen in each interval leading to large variability in the estimator. If the subinterval lengths are too large, interesting features, for example trends and cycles, in the data may be missed.

Kuhl and Wilson (2000) investigate least squares methods for fitting the integrate rate function, $\Lambda(t)$, with rate function of the EPTMP form, as in Equation (2.2.1).

The idea of quantifying the error caused by input modelling that propagates through a simulation model to the simulation output performance measures is now introduced.

2.3 Error Caused by Input Modelling

The "stochastic" in stochastic simulation is a reflection of the input models that are used to drive a simulation through time. Input modelling, as discussed in Chapter 2.2, allows us to form representations of the inputs to a simulation model; this may, for example, be in the form of a statistical distribution, empirical distribution or statistical process. These representations are formed using observations of the system of interest and thus, since only a finite number of observations can ever be collected, contain error. In this thesis the error that propagates to the simulation output due to there being error in the input distributions that drive the simulation is referred to as the error caused by input modelling.

Types of error within simulation include; stochastic estimation error (SEE), the error arising from the generation of random variates during the simulation and model error, caused by differences between the real-world system and the simulation model of it. Here model error includes input modelling error which is caused by having only a finite number of observations to build the input models that drive the simulation. Validation is the process of checking that a simulation model truthfully represents the system it is mimicing. It is not the topic of this thesis, but for more information on validation and verification techniques in simulation see Banks et al. (2013) and references therein. Quantification and reduction of SEE has been well studied, see Nelson (2013) and references therein. Whilst estimating SEE is not the main focus of this thesis it will be discussed when necessary in the following chapters. SEE is known to decrease as the number of replications of a simulation model is increased; error caused by input modelling does not.

In practice, quantification of the error in simulation responses has mainly been restricted to quantifying the SEE. Barton (2012) warns of the danger of not considering error caused by input modelling and Lin et al. (2015) show that error caused by input modelling can be many orders of magnitude larger than SEE. Ignoring error caused by input modelling can lead to over-confidence in the output of the simulation especially when there is little data from which to estimate the input distributions and a large amount of simulation effort has been spent on reducing the SEE.

The mean squared error (MSE) error caused by input modelling can be broken down into variance, known in the literature as input uncertainty (IU), and the squared bias caused by input modelling

$$\text{MSE} = \text{Input Uncertainty} + \text{Bias}^2.$$

The methodology behind input uncertainty quantification and bias caused by input modelling shall now be considered.

2.3.1 Input Uncertainty

First let us formally introduce and define input uncertainty. For simplicity, consider a simulation model with a single input, with true distribution F^c, from which i.i.d data $X_1, X_2, \ldots, X_m$ has been sampled. The true distribution, F^c, from which these values were sampled is unknown, it is therefore estimated by fitting distribution $\widehat{F}$, a function of the observed data. In practice $\widehat{F}$ is used to drive the simulation model; let the observed output of the simulation in replication j be denoted

$$Y_j(\widehat{F}) = \eta(\widehat{F}) + \varepsilon_j(\widehat{F})$$

where $\eta(\widehat{F})$ is the expected simulation response dependent on the estimated distribution $\widehat{F}$ and $\varepsilon_1(\widehat{F}), \varepsilon_2(\widehat{F}), \ldots, \varepsilon_n(\widehat{F})$ are i.i.d random variables representing the noise from replication to replication of the simulation with mean 0 and variance σ^2. In practice this noise is a consequence of the input distributions, caused by the use of random numbers within the simulation to, for example, generate event times. In running the simulation, our interest is in estimating the expected simulation response, $\eta(\widehat{F})$, which may be estimated by taking the average of the simulation output over n replications

$$\bar{Y}(\widehat{F}) = \frac{1}{n}\sum_{j=1}^{n} Y_j = \frac{1}{n}\sum_{j=1}^{n}\left(\eta(\widehat{F}) + \varepsilon_j(\widehat{F})\right) = \eta(\widehat{F}) + \frac{1}{n}\sum_{j=1}^{n}\varepsilon_j(\widehat{F}).$$

As the number of replications, n, gets large the noise in the simulation is driven down to 0.

The variability in $\bar{Y}(\widehat{F})$ can be broken down, using the total law of variance, into input uncertainty and stochastic estimation error as follows

$$\operatorname{Var}(\bar{Y}(\widehat{F})) = \operatorname{Var}[\operatorname{E}(\bar{Y}(\widehat{F})|\widehat{F})] + \operatorname{E}[\operatorname{Var}(\bar{Y}(\widehat{F})|\widehat{F})]. \tag{2.3.1}$$

Input uncertainty, the first term in Equation (2.3.1), is the variability in $\bar{Y}(\widehat{F})$ that comes from having estimated the input distributions. Since the fitted distribution, $\widehat{F}$, is based on real-world data it is independent of the noise in the simulation and thus IU reduces to

$$\text{IU} = \operatorname{Var}[\eta(\widehat{F})]. \tag{2.3.2}$$

The second term in Equation (2.3.1) represents the SEE that arises in the simulation model, it can be estimated using the sample variance of the simulation response, S^2/n.

There exist both parametric and non-parametric IU quantification techniques. Non-parametric methods focus on the statistical technique of bootstrap resampling and parametric techniques can be split into frequentist and Bayesian methodologies. The different approaches to IU quantification will now be discussed alongside the current literature in this area. Note that the techniques presented here only consider IU quantification in simulation models with homogeneous input distributions. In this thesis when we say homogeneous we mean homogeneous with respect to time. In Chapter 3 new methodology for input uncertainty quantification in simulation models with nonhomogeneous inputs is presented.

Barton and Schruben (1993, 2001) consider three approaches to input uncertainty quantification. The first is known as direct resampling and is based on the variability in the mean simulation response having used a different sample of data in each replication. In reality it is not guaranteed that multiple real-world samples will be available; a solution to this is to use bootstrapping, see Efron and Tibshirani (1986), which is the basis for their second approach. Bootstrapping mimics the effect of having multiple real-world samples by either sampling from the original data with replacement or generating new samples from the estimate fitted input distribution. Barton and Schruben (2001) present a bootstrap resampling method for IU quantification working from empirical input distributions. They also present a third approach of IU quantification based on randomly changing the increments of the empirical distribution function within each replication.

Ankenman and Nelson (2012) provide a quick method for assessing the impact of input uncertainty on simulation performance which requires relatively little additional simulation effort having run the simulation to gain the output of interest. This method is based on a random-effects model. The random effects model assumes multiple samples of size m have been observed and thus there are multiple estimates of the input model $\widehat{F_i}$, for $i = 1,2,\ldots,b$. In reality it may not be the case that multiple samples have been observed, bootstrapping is therefore utilised to mimic having observed the additional samples of size m. Their estimator is a measure of the difference between an estimate of the total variability of the simulation output and the SEE as, intuitively, the difference can be attributed to input uncertainty. This estimator may be crude but Ankenman and Nelson (2012) also provide a method for assessing which inputs to the simulation are the largest contributors to input uncertainty which can be a good source of information for follow up data collection. One drawback of the proposed follow up experiment is its complexity. Song and Nelson (2013) provide a follow-up analysis that requires no additional simulation experiments and provides more information than the method of Ankenman and Nelson (2012).

Barton et al. (2013) present a metamodel assisted bootstrapping method for constructing a confidence interval about the mean response of a system that, unlike traditional simulation confidence intervals, takes into account both SEE and IU. The metamodel of the mean response is built using experimental design and leads to computational savings as input uncertainty can be propagated through the metamodel to the response without simulating. Similarly, Barton et al. (2010) and Xie et al. (2014b, 2016) present meta-model assisted bootstrapping frameworks for IU quantification in stochastic simulation models with dependent input models where IU also arises in the estimation of the correlation matrix. Within the proposed methods a stochastic kriging meta-model is used to propagate IU to the mean response.

Song and Nelson (2015) also adopt a meta-model approach within their IU quantification method by introducing a mean-variance effects model. This treats the mean response as a function of the means and variances of the input distributions. For a simulation model with a single input distribution this is

$$\eta(\widehat{F}) = \beta_0 + \beta_1 \mu(\widehat{F}) + \nu\sigma^2(\widehat{F}), \tag{2.3.3}$$

where $\mu(\widehat{F})$ and $\sigma^2(\widehat{F})$ represent the mean and variance of the input distribution $\widehat{F}$, and β_0, β_1 and ν are constant coefficients to be estimated via least squares regression. Given Model (2.3.3) input uncertainty is approximated by

$$\mathrm{Var}\,[\eta(\widehat{F})] = \beta_1^2 \mathrm{Var}\,[\mu(\widehat{F})] + \nu^2 \mathrm{Var}\,[\sigma^2(\widehat{F})] + 2\beta_1\nu\mathrm{Cov}[\mu(\widehat{F}), \sigma^2(\widehat{F}))].$$

Bootstrap sampling is utilised to fit the mean-variance effects model. The method allows consideration of both parametric and empirical inputs to the simulation model, see Song et al. (2014) for further details. In Chapter 3 more detail is provided and this method is extended to simulation models with piecewise-constant NHPP arrival processes.

In Chapter 3 the method of Cheng and Holland (1998) is also extended. They use a Taylor series approach to enable input uncertainty quantification in simulation models with parametric input distributions. Note that, by only considering parametric distributions, input uncertainty is just parameter uncertainty. Their method takes a first-order Taylor series approximation of the expected simulation response

$$\eta(\widehat{\boldsymbol{\theta}}) \approx \eta(\boldsymbol{\theta}^c) + \nabla\eta(\boldsymbol{\theta}^c)(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^c)^T, \tag{2.3.4}$$

where $\boldsymbol{\theta}^c$ denotes the vector of true input parameters and $\widehat{\boldsymbol{\theta}}$ is the vector of maximum likelihood estimators (MLEs) of the input parameters given the observed data. Taking the variance of (2.3.4) gives the estimate of IU,

$$\mathrm{Var}\left[\eta(\widehat{\boldsymbol{\theta}})\right] \approx \nabla\eta(\boldsymbol{\theta}^c)\mathrm{Var}(\widehat{\boldsymbol{\theta}})\nabla\eta(\boldsymbol{\theta}^c)^T,$$

which is asymptotically correct as the amount of input data goes to infinity. Within this Taylor series approach the gradient term, $\nabla\eta(\boldsymbol{\theta}^c)$, must be estimated as $\boldsymbol{\theta}^c$ is unknown. Lin et al. (2015) present the internal gradient estimator of Wieland and Schmeiser (2006) which allows calculation of $\nabla\eta(\boldsymbol{\theta}^c)$ with no additional simulation effort. Cheng and Holland (2004) use a Taylor series approach to provide a confidence interval that takes into account both SEE and parameter uncertainty.

Turning now to the Bayesian approaches for input uncertainty quantification, Biller and Corlu (2011) also focus on the construction of confidence intervals that take into account parameter uncertainty, with specific focus on the parameters of correlated normal-to-anything (NORTA) distributions within large-scale stochastic simulations. Bayesian approaches to IU quantification usually aim to quantify the uncertainty in the choice of distributional family used to represent an input in addition to the uncertainty arising in estimating its parameters. Chick (1997) proposes a Bayesian framework for analysing the output of a simulated system that infers the full distribution of the simulation output including uncertainty from parameter estimates. Chick (2001) presents a Bayesian model averaging approach to input uncertainty quantification which randomly samples an input model and its parameters for use in each replication. Zouaoui and Wilson (2003, 2004) take a similar approach sampling the input parameters from their posterior distributions and estimating the model uncertainty by weighting the simulation results using the posterior model probabilities. Ng and Chick (2001, 2006) suggest sampling plans for reducing parameter uncertainty and thus uncertainty about the expected simulation response.

Xie et al. (2014a) present a fully Bayesian method for measuring the overall uncertainty in the mean response while simultaneously reducing SEE. They use the posterior distribution of the mean simulation response to quantify the overall uncertainty in the mean response and suggest summarising the uncertainty using a credible interval for the expected simulation response. This credible interval can easily be broken down into SEE and IU.

2.3.2 Bias caused by input modelling

Bias caused by input modelling, denoted b, describes how far, on average, the simulation response is from the real-world performance given the error that arises when estimating the input models. In this thesis bias caused by input modelling is considered for simulation models with parametric inputs where the true input parameters, $\boldsymbol{\theta}^c$, are estimated by the maximum likelihood estimators (MLEs), $\boldsymbol{\theta}^{mle}$. There is currently no literature on quantifying the bias caused by input modelling for simulation models with non-parametric inputs. Here the output of the simulation in replication j is

$$Y_j(\boldsymbol{\theta}^{mle}) = \eta(\boldsymbol{\theta}^{mle}) + \varepsilon_j(\boldsymbol{\theta}^{mle}).$$

Bias caused by input modelling arises within the mean simulation response, $\eta(\boldsymbol{\theta}^{mle})$, it is defined by

$$b = \mathrm{E}[\eta(\boldsymbol{\theta}^{mle})] - \eta(\boldsymbol{\theta}^c),$$

where expectation is taken with respect to the sampling distribution of $\boldsymbol{\theta}^{mle}$. This form of bias arises when the response of interest is a non-linear function of its inputs, as is commonly the case in the complex systems for which simulation is used.

When one refers to quantifying the ‘bias’ it is typically the bias of an estimator of a population parameter given a sample of data, averaged over the distribution of possible samples. In our computer-simulation context this bias is also averaged over the natural noise due to generating samples of the stochastic inputs. Stated differently, our estimator is a function of both real-world and simulated sampling. In Chapter 4 we present new methodology for the estimation of bias caused by input modelling is presented along with a bias detection test for assessing when this error is relevant in terms of the total mean squared error (MSE) caused by input modelling.

Standard methods used to estimate bias are the jackknife and the bootstrap method. The jackknife is often considered the go-to choice for bias reduction. Given k independent observations used to estimate population parameter $\widehat{\theta}$; the jackknife estimate of the bias of estimator $\widehat{\theta}$, $\widehat{b}_{JK}$, is

$$\widehat{b}_{JK} = (n-1)\left(\frac{1}{n}\sum_{i=1}^{n}\widehat{\theta}_i - \widehat{\theta}\right)$$

where $\widehat{\theta}_i$ is the estimator calculated from all but the i^{th} data point, referred to as the i^{th} "leave-one-out" estimator. In words, the jackknife is the average of the deviations of each leave-one-out subsample estimator from $\widehat{\theta}$. This bias estimate is correct up to second-order; see Efron (1982). The jackknife estimate of bias can also be used to give a bias-corrected estimator

$$\widehat{\theta}_{JK} = \widehat{\theta} - \widehat{b}_{JK} = n\widehat{\theta} - \frac{(n-1)}{n}\sum_{i=1}^{n}\widehat{\theta}_i.$$

The bootstrap estimator of bias, $\widehat{b}_{BS}$, mimics the collection of repeated samples of data by sampling from the original data with replacement to gain a bootstrap sample of the same length. Lets say b bootstrap samples are collected from the original data, of length n, then the bootstrap estimator of bias is

$$\widehat{b}_{BS} = \frac{1}{b}\sum_{i=1}^{b}\widehat{\theta}_i^{\star} - \widehat{\theta}$$

where $\widehat{\theta}_i^{\star}$ is the estimator calculated from the i^{th} bootstrap sample. In words, the bootstrap estimator of bias measures how far the bootstrap estimators deviate from the original estimator on average, see Efron and Tibshirani (1994). Like the jackknife, the bootstrap estimator of bias can be used to provide a bias correction

$$\widehat{\theta}_{BS} = \widehat{\theta} - \widehat{b}_{BS}.$$

Both the jackknife and bootstrap estimators of bias are widely used non-parametric techniques for bias estimation and reduction. However, neither method is appropriate in the presence of noise due to the amount of simulation effort required to drive the error to zero, see Chapter 4. In Chapter 4, in the presence of noise, a delta approximation of bias, based on a second-order Taylor series expansion, is used for estimation of bias caused by input modelling. The delta method is introduced and utilised in Chapter 4.

2.4 Spline Functions

In this thesis a piecewise polynomial function that is, by construction, continuous and *e* times continuously differentiable will be referred to as an *e* degree spline function. Known uses for spline functions include: interpolation of data, solving differential equations and curve approximation (de Boor, 1978).

In this thesis the interest in spline functions comes from an input modelling perspective. In Chapter 5 a spline-based input model is presented that uses a spline function to represent the arrival rate function of a NHPP.

Interpolation of observations using splines in the presence of exact data, data without noise, has been studied extensively, see de Boor (1978), Shikin and Plis (1995) and references therein. Of course, in the context of simulation and simulating real-world systems, in reality, exact data is rarely available; instead observations are collected from an underlying process in the presence of noise. Here interpolation would model the noise in the model instead of the underlying process of interest, thus some compromise between staying close to the observed data and obtaining a smooth representation must be reached. Smoothing splines were designed for this problem, see Whittaker (1922), Schoenberg (1964), Reinsch (1967) and Eliers and Marx (1996) for the foundations of work in this area. Smoothing splines use least squares methodology to fit the spline in the presence of some penalty on the smoothness of the resulting function. More recently penalised likelihood approaches have also been used as a tool to give a smooth spline representation (Gray, 1992); this approach is built upon in Chapter 5. The composition of spline functions as a linear combination of B-spline basis functions is now discussed.

2.4.1 Basis functions and spline functions

A e-degree spline function can be described as a linear combination of n, e-degree basis splines, otherwise known as B-splines. Let the k^{th} e-degree B-spline be denoted by $B_{k,e,\boldsymbol{s}_k}(\cdot)$, $k = 1,2,\ldots,n$, where $\boldsymbol{s}_k = \{s_{k-(e+1)}, s_{k-e}, \ldots, s_k\}$ is the knot sequence over which the B-spline is defined. A spline function is therefore denoted by

$$\lambda(t) = \sum_{k=1}^{n} c_k B_{k,e,\boldsymbol{s}_k}(t), \tag{2.4.1}$$

where c_k is the k^{th} spline function coefficient, $k = 1,2,\ldots,n$. All B-splines are defined over knot sequences. Given $\boldsymbol{s}_k$, the k^{th} B-spline has the following properties:

- Local support; $B_{k,e,\boldsymbol{s}_k}(t) > 0$ for $t \in (s_{k-(e+1)}, s_k)$ only.
- Positivity; $B_e(t) \geq 0$ for all t.

For $e > 1$, B-splines can be composed recursively from lower degree B-splines using the following recurrence relation

$$B_{k,e,\boldsymbol{s}_k}(t) = \frac{t - s_{k-(e+1)}}{s_{k-1} - s_{k-(e+1)}} B_{k,e-1,\boldsymbol{s}_k}(t) + \frac{s_k - t}{s_k - s_{k-e}} B_{k+1,e-1,\boldsymbol{s}_{k+1}}(t), \tag{2.4.2}$$

for $t \in [s_{k-(e+1)}, s_k)$; at the lowest level this is

$$B_{k,0,\boldsymbol{s}}(x) = \begin{cases} 1 & \text{if } s_{k-1} \leq x < s_k \\ 0 & \text{otherwise,} \end{cases}$$

see de Boor (1978) for the proof. Figure 2.4.1 demonstrates this recursion by illustrating the lower degree component splines that make up a cubic B-spline. For clarity, the four zero degree, B-splines from which the linear B-splines are composed were omitted from the figure.

To compose a spline function from n B-splines the n local knot vectors $\boldsymbol{s}_k$, $k = 1,2,\ldots,n$ are combined. The resulting knot sequence of the spline function, $\boldsymbol{s} = \{s_{-e}, s_{-e+1}, \ldots, s_0, s_1, \ldots, s_{n+1}\}$, has length $n+e+1$. Within this thesis the first knot in the knot sequence will consistently be denoted by s_{-e}, when there is interest in the spline function on the interval $[0,T]$ by setting $s_0 = 0$ and $s_{n-e} = T$ this means $e+1$ B-splines are non-zero for all $t \in [0,T]$. Figure 2.4.2 illustrates this when the interval of interest is $[0,3]$. It shows the support of 7 B-splines on uniform knot vector $\boldsymbol{s} = \{-3,-2,\ldots,7\}$. Spline functions constructed from B-splines on uniform knot sequences are known as cardinal splines. In general, the knots of a spline function need not be uniformly spaced; other common knot vectors space the

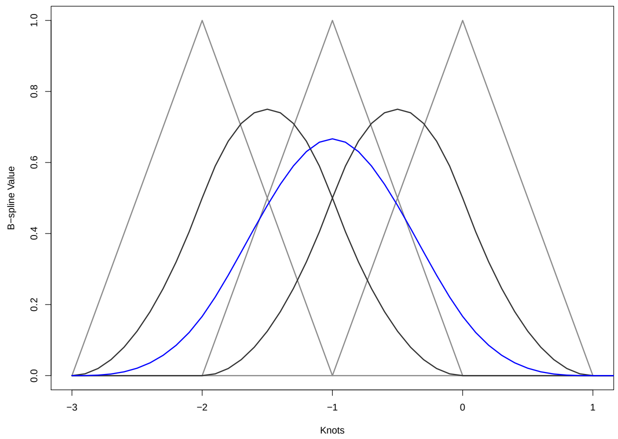


Figure 2.4.1: A single cubic B-spline (blue) and the lower degree, quadratic (dark grey) and linear (light grey), B-splines from which it is composed.

knots such that around the same number of observations fall between each knot, see Gray (1992). But cardinal splines come with certain advantages since there is essentially a single B-spline in use and all other B-splines are horizontal translations of the first.

The definition of a spline function of degree d with knot sequence $\boldsymbol{s}$ is any linear combination of B-splines of order e for the knot sequence $\boldsymbol{s}$. Let the collection of all such functions be denoted by $\mho_{e,\boldsymbol{s}}$ where,

$$\mho_{e,\boldsymbol{s}} = \left\{ \sum_k c_k B_{k,e,\boldsymbol{s}}(t) : c_k \ \in \ \mathbb{R} \, \forall \, k \right\}.$$

Note that, by construction using recurrence relation (2.4.2), as presented by de Boor (1978), a spline function is continuous and e times continuously differentiable. Once the $n+e+1$ knots have been placed the value of the n B-splines is fixed for all t. The shape of the spline function $\lambda(t;\boldsymbol{c})$ is therefore controlled by the value of the spline coefficients $\boldsymbol{c} = (c_k)_{k=1}^{n}$.

Figure 2.4.3 illustrates two cubic spline functions on the same knot vector with different spline coefficients, $\boldsymbol{c} = (c_k)_{k=1}^{n}$, along with the B-spline basis functions used in their composition and markers to show the interval, [0,3], in which $e+1$ B-splines are active. As the number of knots, n, is increased the number of B-splines used to compose the spline function, gets larger. This gives increasing flexibility of the shape of the spline function.

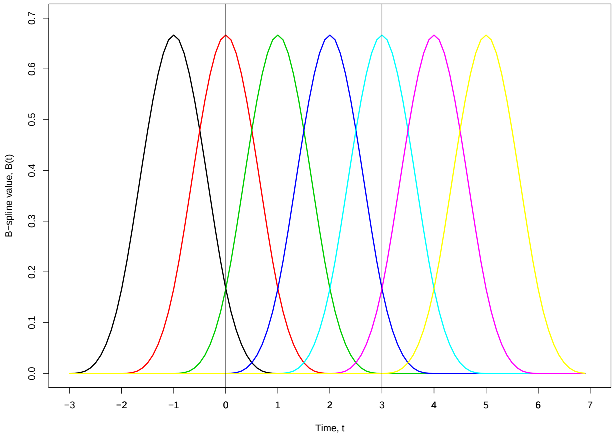


Figure 2.4.2: The support of $n = 7$ B-splines over the interval $[0,3)$ given uniform knot vector $\boldsymbol{s} = \{-3,-2,-1,0,1,2,3,4,5,6,7\}$

Figure 2.4.4 illustrates two spline functions with double the number of knots as those in Figure 2.4.3; it is clear that these spline functions are more flexible.

The topic of spline functions is returned to in Chapter 5 where a spline function is used to estimate, and generate event times from, the arrival rate function of a NHPP.

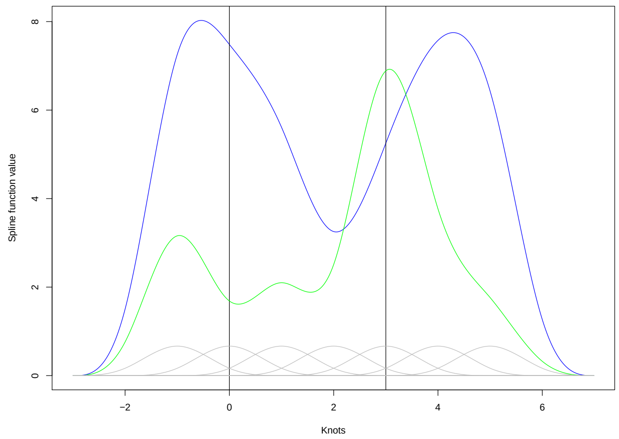


Figure 2.4.3: Two cubic spline functions (green and blue) on knot vector $\boldsymbol{s} = \{-3,\ -2,\ -1,\ 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\}$ with spline coefficients $\boldsymbol{c}_1 = \{9.02, 7.46, 6.04, 2.04, 5.34, 8.13, 7.61\}$ (blue) and $\boldsymbol{c}_2 = \{4.57, 0.69, 2.80, 0.70, 9.45, 2.80, 1.94\}$ (green) along with the B-spline basis functions from which they were composed (grey).

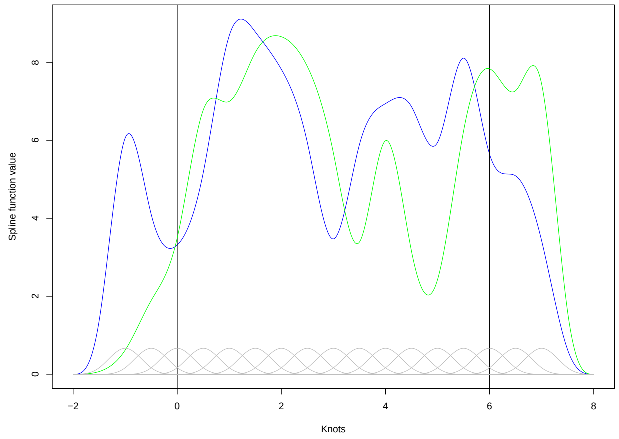


Figure 2.4.4: Two spline functions (green and blue) on knot vector $\boldsymbol{s} = \{-2.0,\ -1.5,\ -1.0,\ -0.5,\ 0.0,\ 0.5,\ 1.0,\ 1.5,\ 2.0,\ 2.5,\ 3.0,\ 3.5,\ 4.0,\ 4.5,\ 5.0, 5.5,\ 6.0,\ 6.5,\ 7.0,\ 7.5,\ 8.0\}$ along with the B-spline basis functions from which they were composed (grey).

Input Uncertainty Quantification for Simulation Models with Piecewise-constant Non-stationary Poisson Arrival Processes

3.1 Introduction

Within simulation models, more often than not, the true input models used to drive the system are unknown. When observations are available from the system of interest the input models can be estimated, and this causes uncertainty to arise within the simulation output. This error is known as input uncertainty (IU). Overlooking IU is still a common error in the simulation community where practitioners treat the estimated input models as correct. This can be risky, particularly if the sample of real-world data is small, and could result in misleading outputs. The survey by Barton (2012) showed that in some cases input uncertainty overwhelms stochastic estimation error, the error arising from the generation of random variates during the simulation; it should, therefore, not be ignored.

Recently input uncertainty techniques have been implemented in the commercial software Simio (Simio LLC) making it easier for simulation users to quantify the effect of input uncertainty without having to manually implement a complex statistical procedure. However, this software is limited to i.i.d processes. For a review of input uncertainty quantification techniques see the survey papers by Barton (2012) or Song et al. (2014).

In operational research most simulation models have some form of arrival process. Examples include call centers, supply chains or accident and emergency departments where customers or demand can occur according to either a stationary or non-stationary arrival process. Input uncertainty for nonhomogeneous arrival processes is yet to be addressed. This chapter aims to fill this gap by quantifying input uncertainty in simulation models with piecewise-constant, nonhomogeneous Poisson arrival processes. Piecewise-constant arrival rate functions are often used in practice in simulation studies as they provide flexibility and are conveniently fit to count data. They are included in many software packages such as Simio (Simio LLC), SIMUL8 (Simul8 Corporation) and Arena (Rockwell Automation). It is therefore a natural step to want to quantify the uncertainty propagated to the simulation output due to the estimation of nonhomogeneous arrival processes. We extend two existing methods for quantifying IU due to i.i.d. input processes to cover nonhomogeneous Poisson processes with piecewise-constant arrival rates estimated from count data. Further, we improve one method by exploiting the knowledge that the process is Poisson allowing it to handle arrival processes with many rate changes. We also demonstrate how change-point analysis can be used to obtain a parsimonious representation of the piecewise-constant arrival rate function.

The chapter is organised as follows. In §3.2 we present background on current IU quantification techniques and discuss methods for modelling nonhomogeneous Poisson arrival processes. In §3.3 new methods, building on the work of Cheng and Holland (1997) and Song and Nelson (2015) are presented. This is followed by an empirical evaluation and realistic illustration of the methods in §3.4. We finish with conclusions and suggestions for further work in §3.5.

3.2 Background

An early contribution to the IU literature came from Cheng and Holland (1997) who modelled IU using a Taylor series expansion of the mean response as a function of the input distribution parameters. An adaptation of this method was later given by Lin et al. (2015) making use of internal gradient estimation, derived by Wieland and Schmeiser (2006), to reduce quantification of input uncertainty to a single experiment. An alternative approach was given by Song and Nelson (2015) who present a mean-variance effects model for quantifying IU. This method, although not asymptotically justified, makes intuitive sense as the performance measures are likely to depend greatly on the mean and variance of the input distributions.

There are also Bayesian techniques that can be implemented to assist in quantifying uncertainty. Chick (2001) first employed Bayesian techniques enabling the incorporation of prior knowledge of input distributions into simulation modelling. In this method prior information is used for the selection of the input distributions only and input uncertainty is still calculated using the frequentist approach of finding and subtracting the simulation estimation error from the total uncertainty. Zouaoui and Wilson (2010) extended this technique using the posterior probability of the candidate distributions to weight the simulation response but again use frequentist techniques for IU quantification. Recently Xie et al. (2014a) developed a fully Bayesian approach for quantifying uncertainty using Gaussian processes to find the posterior distribution of the simulation performance measure of interest. This is then summarized by a credible interval which can easily be dissected to find an estimate for the input uncertainty.

Modelling nonhomogeneous Poisson arrival processes (NHPPs) is also key to our problem. Using Poisson processes has its advantages: they have good properties that make them easy to simulate using thinning or inversion. Kuhl and Wilson (2009) consider both parametric and non-parametric input model approximations, with respect to NHPPs.

Our focus in this chapter is on count data and we model the rate function, $\lambda(t)$, as a piecewise-constant function over q intervals. The intervals, $(0,t_1],(t_1,t_2],\ldots,(t_{q-1},t_q]$, will represent the intervals over which the rate is unchanged. Chen and Gupta (2011) give a way to identify, from count data, where change points in the rate function occur using hypothesis testing. This technique will be utilised in §3.4 as a pre-processing tool to reduce the number of parameters in our model. Employing piecewise-constant $\lambda(t)$ is justified by Henderson (2003) who showed that asymptotically, increasing the number of observations of a process whilst simultaneously decreasing the interval size leads to the true arrival rate function of interest under mild conditions.

3.3 Methods

Before considering IU quantification for piecewise-constant NHPPs we set up our approach by reviewing two existing techniques for quantifying input uncertainty in simulation models with stationary arrival processes.

For ease of explanation consider the simulation of a single queue with two driving processes. Let the true input distributions be denoted by $\mathbf{F}^c$; in reality these distributions are unknown and therefore estimated distributions $\widehat{\mathbf{F}}$ will be used to drive the simulation. We will assume the arrivals follow a Poisson process, with true rate parameter λ^c, denote this by F_λ. The service distribution, depending on the situation, may be estimated by a parametric or non-parametric distribution but for ease of exposition we treat it as a parametric distribution with true parameter/s $\boldsymbol{\theta}^c$; denote this $\mathrm{F}_{\boldsymbol{\theta}}$. Note that the form of $\mathrm{F}_{\boldsymbol{\theta}}$ will have an effect on the approach we will take. This gives the parameter space $(\lambda^c,\boldsymbol{\theta}^c)$ where $\boldsymbol{\theta}^c$ is a row vector of parameters from the service distribution, and here $\mathbf{F}^c=(\mathrm{F}_\lambda,\mathrm{F}_{\boldsymbol{\theta}})$.

Given real-world data we have independent counts, $N_1,N_2,\ldots,N_{m_\lambda}$ of the arrival process, observed m_λ times over the interval $[0,T)$, and observations $X_1,X_2,\ldots,X_{m_{\boldsymbol{\theta}}}$ of the service process. Therefore $(\lambda^c,\boldsymbol{\theta}^c)$ can be estimated by their maximum likelihood estimators (MLEs) $(\widehat{\lambda},\widehat{\boldsymbol{\theta}})$. For example assuming the arrivals follow a Poisson process implies that the arrival counts can be represented by a Poisson distribution, $N_1,N_2,\ldots,N_{m_\lambda}\sim$

Poisson(λ^cT), and the MLE of the arrival rate is therefore

$$\widehat{\lambda} = \frac{\Sigma_{i=1}^{m_\lambda} N_i}{m_\lambda \mathrm{T}}.$$

This gives the estimated distributions $\mathrm{F}_{\widehat{\lambda}}$ and $\mathrm{F}_{\widehat{\boldsymbol{\theta}}}$ used to drive the simulation. The simulation goal is to estimate $\eta(\lambda^c, \boldsymbol{\theta}^c)$, the expected value of the output of the simulation given the true input parameters. We describe the output from replication j of the simulation by

$$Y_j(\lambda, \boldsymbol{\theta}) = \eta(\lambda, \boldsymbol{\theta}) + \varepsilon_j(\lambda, \boldsymbol{\theta}) \qquad j = 1, 2, \ldots, r$$

where ε represents stochastic noise and has mean 0 and variance $\sigma^2(\lambda, \boldsymbol{\theta})$, and r is the total number of replications. Given the MLEs $(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})$ a nominal performance measure estimate of $\eta(\lambda^c, \boldsymbol{\theta}^c)$ is

$$\bar{Y}(\widehat{\lambda}, \widehat{\boldsymbol{\theta}}) = \frac{\Sigma_{j=1}^{r} Y_j(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})}{r}.$$

This has variance $\mathrm{Var}[\bar{Y}(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})]$ which breaks down into input uncertainty and simulation estimation error. Note that most simulation studies ignore input uncertainty because it is believed to be difficult to quantify. In reality input uncertainty is just the variance of the expected value of the output of the simulation with respect to the estimated parameters $(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})$; this can be denoted by

$$\sigma_I^2 = \mathrm{Var}[\eta(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})] = \mathrm{Var}[\mathrm{E}(Y(\widehat{\lambda}, \widehat{\boldsymbol{\theta}}) | \widehat{\lambda}, \widehat{\boldsymbol{\theta}})].$$

See Chapter 2 for the full derivation. The other contribution to uncertainty in the output of the simulation comes from simulation estimation error caused by the generation of random variates during the simulation. Simulation estimation error is denoted by $\sigma^2(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})/r$ which can be estimated using the sample variance S^2/r.

3.3.1 Cheng and Holland

Cheng and Holland (1997) consider only parametric distributions as inputs to the simulation model. This simplifies input uncertainty to parameter uncertainty. Using a Taylor

Series approximation, if $\eta(\lambda, \boldsymbol{\theta})$ is continuously differentiable then to first order it can be expressed as

$$\eta(\widehat{\lambda}, \widehat{\boldsymbol{\theta}}) \approx \eta(\lambda^c, \boldsymbol{\theta}^c) + \nabla\eta(\lambda^c, \boldsymbol{\theta}^c)((\widehat{\lambda}, \widehat{\boldsymbol{\theta}}) - (\lambda^c, \boldsymbol{\theta}^c))^T$$

where $\nabla\eta(\lambda^c, \boldsymbol{\theta}^c)$ is the gradient of the expected value of the performance measure with respect to the input parameters λ and $\boldsymbol{\theta}$. Input uncertainty, $\text{Var}[\eta(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})]$, can then be approximated by

$$\text{Var}[\eta(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})] \approx \nabla\eta(\lambda^c, \boldsymbol{\theta}^c)\text{Var}(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})\nabla\eta(\lambda^c, \boldsymbol{\theta}^c)^T. \tag{3.3.1}$$

In reality, none of the terms on the right-hand side of Equation (3.3.1) are known and so must be estimated.

If the two input distributions are assumed independent, then $\text{Var}(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})$ can be denoted by

$$\text{Var}(\widehat{\lambda}, \widehat{\boldsymbol{\theta}}) = \begin{pmatrix} \text{Var}(\widehat{\lambda}) & \mathbf{0} \\ \mathbf{0} & \text{Var}(\widehat{\boldsymbol{\theta}}) \end{pmatrix}.$$

This can be estimated by $\widehat{\text{Var}}(\widehat{\lambda}, \widehat{\boldsymbol{\theta}}) = \text{I}^{-1}(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})$, the inverse Fisher information matrix of the MLEs evaluated at $(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})$.

Estimation of the gradient is critical. One method is to use the internal gradient estimator of Wieland and Schmeiser (2006), as seen in Lin et al. (2015). This enables $\nabla\eta(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})$ to be evaluated using no additional simulation effort. We specialise this gradient estimation method to our situation below. Although based on similar ideas, the gradient estimation described here is distinct from the Taylor series expansion of $\eta(\hat{\lambda}, \hat{\boldsymbol{\theta}})$ employed by Cheng and Holland (1997).

To ease understanding of how we estimate the gradient, consider a simulation model with a single input distribution. Let this describe arrivals to a system and be approximated by real-world data where arrival count observations $N_1, N_2, \dots, N_{m_\lambda} \sim \text{Poisson}(\lambda^c\text{T})$. We assume arrivals are simulated over the full interval $[0, T)$. From these observations $\widehat{\lambda}$ can be found, this is then, for the purpose of the internal gradient estimation method, considered to be the true arrival rate λ^c. In replication j, the rate $\widehat{\lambda}$ is used to drive the simulation and the count of the number of simulated arrivals in the interval is recorded. Denote this by d_j

for replication j. This count can then be used to re-estimate the arrival rate; we call this estimate $\bar{\lambda}_j$ where $\bar{\lambda}_j = d_j/T$. Note that we assume $\mathrm{E}[\bar{\lambda}_j] = \widehat{\lambda}$ since the parameter $\widehat{\lambda}$ was used to run the simulation over j replications. This results in pairs of observations $(Y_j, \bar{\lambda}_j)$. Assuming the output of the simulation depends on the input models, as is most likely the case, then $(Y_j, \bar{\lambda}_j)$ are expected to be dependent. Moreover, if their joint distribution is assumed to be approximately bivariate normal then

$$\mathrm{E}[Y_j(\widehat{\lambda})|\bar{\lambda}_j] = \eta(\widehat{\lambda}) + \Sigma_{Y\bar{\lambda}}\Sigma^{-1}_{\bar{\lambda}\bar{\lambda}}(\bar{\lambda}_j - \widehat{\lambda}) = \delta_0 + \delta_1\bar{\lambda}_j$$

where $\Sigma_{Y\bar{\lambda}}$ is the covariance between Y_j and $\bar{\lambda}_j$ and $\Sigma_{\bar{\lambda}\bar{\lambda}}$ is the variance of $\bar{\lambda}_j$. Here the derivative of the expected response with respect to λ, the gradient, estimated at $\widehat{\lambda}$ equals $\delta_1 = \Sigma_{Y\bar{\lambda}}\Sigma^{-1}_{\bar{\lambda}\bar{\lambda}}$ which can easily be estimated using least squares regression.

This method can be extended when there are multiple input distributions, which is often the case in simulation models. Recall in our simulation model there is an arrival and service distribution. To find $\bar{\boldsymbol{\theta}}$, for the service distribution, this method is just repeated with respect to $\boldsymbol{\theta}$. This gives r independent and identically distributed (i.i.d) vectors $(Y_j, (\bar{\lambda}_j, \bar{\boldsymbol{\theta}}_j))$, $j = 1, 2, \ldots, r$.

Lin et al. (2015) suggest the joint distribution of $(Y_j, (\bar{\lambda}_j, \bar{\boldsymbol{\theta}}_j))$ should now be considered multivariate normal, a natural extension of the previous approach, which gives

$$\mathrm{E}[Y_j(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})|(\bar{\lambda}_j, \bar{\boldsymbol{\theta}}_j)] = \eta(\widehat{\lambda}, \widehat{\boldsymbol{\theta}}) + \Sigma_{Y(\bar{\lambda},\bar{\boldsymbol{\theta}})}\Sigma^{-1}_{(\bar{\lambda},\bar{\boldsymbol{\theta}})(\bar{\lambda},\bar{\boldsymbol{\theta}})}\left((\bar{\lambda}_j, \bar{\boldsymbol{\theta}}_j) - (\widehat{\lambda}, \widehat{\boldsymbol{\theta}})\right)^T = \delta_0 + \boldsymbol{\delta}_1(\bar{\lambda}_j, \bar{\boldsymbol{\theta}}_j)^T.$$

The gradient of $\nabla\eta(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})$ is $\boldsymbol{\delta}_1$ which, again, can be obtained by least squares regression. We now have estimates of both $\mathrm{Var}(\widehat{\lambda}, \widehat{\boldsymbol{\theta}})$ and $\nabla\eta(\lambda^c, \boldsymbol{\theta}^c)$ and can therefore quantify IU using Equation (3.3.1).

3.3.2 Song and Nelson

Song and Nelson (2015) suggest a different approximation of the mean function. Their approach is applicable to both parametric and non-parametric distributions, unlike the approach of Cheng and Holland (1997). We therefore let the output of the j^{th} replication of the simulation, given the collection of input distributions $\widehat{\mathbf{F}}$, be denoted by $Y_j(\widehat{\mathbf{F}}) = \mathrm{E}[Y(\widehat{\mathbf{F}})|\widehat{\mathbf{F}}] + \varepsilon_j$, where the distribution of ε_j could depend on $\widehat{\mathbf{F}}$.

They assume that the output mean $\mathrm{E}[Y(\widehat{\mathbf{F}})|\widehat{\mathbf{F}}]$ can be represented as a function of the mean, $\mu(\widehat{\mathbf{F}})$, and variance, $\sigma^2(\widehat{\mathbf{F}})$, of the input distributions alone. Since $\mathrm{E}[Y(\widehat{\mathbf{F}})|\widehat{\mathbf{F}}]$ is a random variable dependent on $\widehat{\mathbf{F}}$ it can be thought of as a function, $\eta(\widehat{\mathbf{F}})$, which, in the case of our queueing illustration, Song and Nelson (2015) approximate as

$$\eta(\widehat{\mathbf{F}}) \approx \beta_0 + \beta_\lambda\ \mu(F_{\widehat{\lambda}}) + \nu_\lambda\ \sigma^2(F_{\widehat{\lambda}}) + \beta_{\boldsymbol{\theta}}\ \mu(F_{\widehat{\boldsymbol{\theta}}}) + \nu_{\boldsymbol{\theta}}\ \sigma^2(F_{\widehat{\boldsymbol{\theta}}}).$$

This is called a mean-variance effects model, and it can be extended to any number of stationary input distributions.

Song and Nelson (2015) fit this model by generating B bootstrap samples from $\widehat{\mathbf{F}}$, then using the empirical distribution of these bootstrap samples, $\widehat{\mathbf{F}}^{\star}$, to drive B simulations. Empirical distributions are used to obviate the need to refit a parametric distribution to each bootstrap sample from $\widehat{\mathbf{F}}$, and because it makes certain variance and covariance terms (see below) easier to compute.

Consider our assumption that the observed arrival counts follow a Poisson process with true rate λT. Here the mean and variance of the observed counts, $\mu(\widehat{F_\lambda})$ and $\sigma^2(\widehat{F_\lambda})$, both equal $\widehat{\lambda}$T, simplifying the mean-variance model. Note that Song and Nelson (2015) did not consider the use of counts to estimate the arrival rate, as we do here. Now only one regression coefficient is needed to represent the arrival process

$$\eta(\widehat{\mathbf{F}}) \approx \beta_0 + \beta_\lambda\ \mu(F_{\widehat{\lambda}}) + \beta_{\boldsymbol{\theta}}\ \mu(F_{\widehat{\boldsymbol{\theta}}}) + \nu_{\boldsymbol{\theta}}\ \sigma^2(F_{\widehat{\boldsymbol{\theta}}}). \tag{3.3.2}$$

In addition, in this case the bootstrap samples are easy to fit to a Poisson process using the MLE, $\widehat{\lambda}$. Therefore, the bootstrap simulations can be driven by Poisson processes rather than empirical distributions; this makes the method more accurate. These two insights are key to our approach.

From Equation (3.3.2) we derive input uncertainty, σ_I^2,

$$\begin{aligned}\mathrm{Var}[\eta(\widehat{\mathbf{F}})] &= \mathrm{Var}[\beta_0+\beta_\lambda\ \mu(F_{\widehat{\lambda}})+\beta_{\boldsymbol{\theta}}\ \mu(\widehat{F}_{\boldsymbol{\theta}})+v_\theta\ \sigma^2(\widehat{F}_{\boldsymbol{\theta}})],\\ &= \beta_\lambda^2\ \mathrm{Var}[\mu(F_{\widehat{\lambda}})]+\beta_{\boldsymbol{\theta}}^2\mathrm{Var}[\mu(\widehat{F}_{\boldsymbol{\theta}})]+v_{\boldsymbol{\theta}}^2\mathrm{Var}[\sigma^2(\widehat{F}_{\boldsymbol{\theta}})]+2v_{\boldsymbol{\theta}}\beta_{\boldsymbol{\theta}}\ \mathrm{Cov}[\mu(\widehat{F}_{\boldsymbol{\theta}}),\sigma^2(\widehat{F}_{\boldsymbol{\theta}})],\end{aligned} \tag{3.3.3}$$

assuming independence among the input distributions. Expression (3.3.3) can be approximated, through the use of bootstrap sampling, by

$$\mathrm{Var}[\eta(\widehat{\mathbf{F}})] = \mathrm{Var}[\eta(\widehat{\mathbf{F}})|\mathbf{F}^c] \approx \mathrm{Var}[\eta(\widehat{\mathbf{F}}^\star)|\widehat{\mathbf{F}}],$$

where

$$\begin{aligned}\mathrm{Var}[\eta(\widehat{\mathbf{F}}^\star)|\widehat{\mathbf{F}}] =\ & \beta_\lambda^2\ \mathrm{Var}[\mu(F_{\widehat{\lambda}}^\star)|F_{\widehat{\lambda}}]+\beta_{\boldsymbol{\theta}}^2\mathrm{Var}[\mu(\widehat{F}_{\boldsymbol{\theta}}^\star)|\widehat{F}_{\boldsymbol{\theta}}]+v_{\boldsymbol{\theta}}^2\mathrm{Var}[\sigma^2(\widehat{F}_{\boldsymbol{\theta}}^\star)|\widehat{F}_{\boldsymbol{\theta}}]\\ &+2v_{\boldsymbol{\theta}}\beta_{\boldsymbol{\theta}}\ \mathrm{Cov}[\mu(\widehat{F}_{\boldsymbol{\theta}}^\star),\sigma^2(\widehat{F}_{\boldsymbol{\theta}}^\star)|\widehat{F}_{\boldsymbol{\theta}}].\end{aligned}$$

Firstly looking at the arrival distribution, if we let $F_{\widehat{\lambda}}^\star$ denote the Possion distribution fitted by the parametric bootstrap sample of arrival counts, then $\mathrm{Var}[\mu(F_{\widehat{\lambda}}^\star)|F_{\widehat{\lambda}}] = \mathrm{Var}[\widehat{\lambda}^\star|F_{\widehat{\lambda}}] = \widehat{\lambda}/m_\lambda\mathrm{T}$. For the service process, which we will assume to be non-parametric, $\widehat{F}_{\boldsymbol{\theta}}^\star$, $\mu(\widehat{F}_{\boldsymbol{\theta}}^\star)$ and $\sigma^2(\widehat{F}_{\boldsymbol{\theta}}^\star)$ are given by the mean and second sample central moment of the bootstrapped sample $X_1^\star, X_2^\star, \dots X_{m_{\boldsymbol{\theta}}}^\star$. As the number of observations increases this approximation is asymptotically justified and expressions for the variance and covariance can be found by

$$\begin{aligned}\mathrm{Var}[\mu(\widehat{F}_{\boldsymbol{\theta}}^*)|\widehat{F}_{\boldsymbol{\theta}}] &= \frac{M_{\boldsymbol{\theta}}^2}{m_{\boldsymbol{\theta}}}\\ \mathrm{Var}[\sigma^2(\widehat{F}_{\boldsymbol{\theta}}^*)|\widehat{F}_{\boldsymbol{\theta}}] &\approx \frac{M_{\boldsymbol{\theta}}^4-(M_{\boldsymbol{\theta}}^2)^2}{m_{\boldsymbol{\theta}}}\\ \mathrm{Cov}[\mu(\widehat{F}_{\boldsymbol{\theta}}),\sigma^2(\widehat{F}_{\boldsymbol{\theta}}^*)|\widehat{F}_{\boldsymbol{\theta}}] &\approx \frac{M_{\boldsymbol{\theta}}^3}{m_{\boldsymbol{\theta}}}\end{aligned}$$

where $M_{\boldsymbol{\theta}}^k$ is the k^{th} central moment of $\widehat{F}_{\boldsymbol{\theta}}$, and since $\widehat{F}_{\boldsymbol{\theta}}$ is an empirical distribution $M_{\boldsymbol{\theta}}^k = \Sigma_{i=1}^{m_{\boldsymbol{\theta}}}(X_{\boldsymbol{\theta}i}-\bar{X}_{\boldsymbol{\theta}})^k/m_{\boldsymbol{\theta}}$.

To find the coefficients of the mean-variance meta-model the bootstrap experiments are used to fit a regression model which can be used to evaluate $\beta_0, \beta_\lambda, \beta_{\boldsymbol{\theta}}$ and $v_{\boldsymbol{\theta}}$. This gives all components needed to calculate input uncertainty using Equation (3.3.3).

When deciding which method to use in practice, the form of the input distributions is key, as is the amount of data available. Cheng and Holland (1997) require all input distributions to be parametric and therefore the method could be said to have less flexibility. Conversely, Song and Nelson (2015) can handle both parametric and non-parametric distributions but difficulty arises in computing the variance and covariance terms needed to quantify IU for some parametric distributions. Note that in our case we exploit the fact that for Poisson distributions this is easy.

The use of bootstrapping by Song and Nelson (2015) means given any sized sample of observations of either process we should be able to obtain the same approximation of IU. Although be warned that the validity of bootstrapping does come into question for extremely small samples, see Chernick (2008) for a discussion. Unlike the method by Cheng and Holland (1997) which relies on asymptotic theory, and therefore may not give a good approximation of input uncertainty when the number of observations is small. But being asymptotically justified could be seen as an advantage, Song and Nelson (2015) rely on their intuitive model which may not perform well in situations where the output of the simulation cannot be described well by the first two moments of the input distributions.

It will be of interest to see if the strengths and weaknesses of either method translate to cases where nonhomogeneous arrival processes are included in the simulation model; this will be covered in §3.4.

3.3.3 Nonhomogeneous Arrival Processes

We now present two methods for quantifying input uncertainty in simulation models driven using at least one piecewise-constant, nonhomogeneous Poisson arrival process. These methods build upon the work of Cheng and Holland (1997) and Song and Nelson (2015) but introduce the idea of modelling the input arrival distributions using arrival count observations instead of inter-arrival time observations. The assumption that these arrival counts follow a Poisson distribution is key to our new methods and leads to a useful simplification in both cases.

Consider a piecewise-constant NHPP with q distinct arrival rates over the intervals $[0,t_1)$, $[t_1,t_2),\ldots,[t_{q-1},T)$. Each interval can be considered as a single input distribution to the simulation with the observation interval matching the simulation interval. Again let us consider a simple queueing model with a stationary service distribution and an arrival process described by a piecewise-constant NHPP. The parameter space is now $(\lambda_1,\lambda_2,\ldots,\lambda_q,\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is a row vector describing the parameters of the service process.

We start by describing the Taylor series approximation method for quantifying input uncertainty in this situation. Observed arrival counts in each interval are independent implying no dependence between $\widehat{F}_{\lambda_1},\widehat{F}_{\lambda_2},\ldots,\widehat{F}_{\lambda_q}$. Equation (3.3.1) therefore becomes

$$\sigma_I^2 = \mathrm{Var}[\eta(\widehat{\boldsymbol{\lambda}},\widehat{\boldsymbol{\theta}})] \approx \nabla\eta(\boldsymbol{\lambda}^c,\boldsymbol{\theta}^c)\mathrm{Var}(\widehat{\boldsymbol{\lambda}},\widehat{\boldsymbol{\theta}})\nabla\eta(\boldsymbol{\lambda}^c,\boldsymbol{\theta}^c)^T.$$

This requires estimation of the gradient, $\nabla\eta(\boldsymbol{\lambda}^c,\boldsymbol{\theta}^c)$, and variance matrix, $\mathrm{Var}(\widehat{\boldsymbol{\lambda}},\widehat{\boldsymbol{\theta}})$. The independence of the q arrival processes gives the following diagonal variance matrix

$$\begin{pmatrix} \mathrm{Var}(\widehat{\lambda}_1) & 0 & & \ldots & \mathbf{0} \\ 0 & \mathrm{Var}(\widehat{\lambda}_2) & & & \\ & & \vdots & & \\ & & & \mathrm{Var}(\widehat{\lambda}_q) & \mathbf{0} \\ 0 & & & 0 & \mathrm{Var}(\widehat{\boldsymbol{\theta}}) \end{pmatrix}.$$

Since the arrival counts are assumed to be Poisson, closed form-equations exist for each $\mathrm{Var}(\widehat{\lambda}_i)$, $i=1,2,\ldots,q$. Gradient estimation is also no harder in the nonhomogeneous case using the internal gradient estimation method of Lin et al. (2015). This requires evaluation of $\bar{\lambda}_i$, for $i=1,2,\ldots,q$ and least squares regression of Y_j with respect to the parameter space $(\bar{\boldsymbol{\lambda}}_j,\bar{\boldsymbol{\theta}}_j)$. One concern with this approach is the validity of the first-order approximation if q becomes large over many short intervals. A possible way around this would be to merge small intervals with similar arrival rates using change-point analysis within the pre-processing stage of the experiment; this idea is explored further in §3.4.

Our second method, to be referred to as the mean-variance approximation, makes use of a mean-variance effects model in the same way as Song and Nelson (2015) but uses arrival counts to model the input arrival distribution instead of inter-arrival times. Again we consider each interval of the arrival process as a distinct distribution, each with arrival rate λ_i, for $i = 1,2,\ldots,q$. Assuming the arrival counts follow a Poisson process means $\mu(F_{\widehat{\lambda}_i}) = \sigma^2(F_{\widehat{\lambda}_i})$ for $i = 1,2,\ldots,q$, as seen in §3.3.2, allowing a simplification of the mean-variance effects model. The arrival process therefore only contributes q elements, $\mu(F_{\widehat{\lambda}_i})$ for $i = 1,2,\ldots,q$, to the mean-variance effects model, rather than $2q$. This is a significant simplification when there are many intervals. Formulae exist for both $\mu(F_{\widehat{\lambda}_i})$ and $\text{Var}[\mu(F_{\widehat{\lambda}_i})]$ making the method simple to implement.

We have presented two techniques for approximately quantifying input uncertainty in simulation models with piecewise-constant nonhomogeneous Poisson input processes. However, it may also be of interest to determine the overall contribution of the arrival process to IU to evaluate whether it overwhelms the uncertainty contribution from other input distributions or whether there is a specific interval that contributes substantially to the total IU. Similarly in a simulation model with L input distributions it would be useful to establish the relative contribution of the l^{th} input distribution to input uncertainty as this can be used to indicate where more data should be collected if follow-up analysis were to be carried out.

When the input distribution is stationary, Lin et al. (2015) and Song and Nelson (2015) give ways to approximate the contribution of the l^{th} input model. These techniques can also be used alongside our two new methods for finding the contribution to IU of the arrival process. Consider the i^{th} interval of the arrival process, $F_{\widehat{\lambda}_i}$. Using a Taylor series expansion its contribution $c_{\widehat{\lambda}_i}(m_{\widehat{\lambda}_i})$, is given by

$$c_{\widehat{\lambda}_i}(m_{\widehat{\lambda}_i}) = \nabla\eta(\widehat{\lambda}_i)\widehat{\text{Var}}(\widehat{\lambda}_i)\nabla\eta(\widehat{\lambda}_i)^T,$$

and when the mean-variance approximation method is used this translates to

$$c_{\widehat{\lambda}_i}(m_{\widehat{\lambda}_i}) = \beta^2_{\widehat{\lambda}_i}\text{Var}[\mu(F_{\widehat{\lambda}_i})].$$

Now if we were interested in finding the total contribution of the arrival process this is just the sum of the contributions of the q individual intervals

$$c_{\boldsymbol{\lambda}}(m_{\boldsymbol{\lambda}}) = c_{\lambda_1}(m_{\lambda_1}) + c_{\lambda_2}(m_{\lambda_2}) + \cdots + c_{\lambda_q}(m_{\lambda_q}).$$

Whichever approach is used to quantify input uncertainty, an approximation of the contribution of the l^{th} input model to input uncertainty can be found. From here quantifying the relative contribution of the l^{th} input distribution, $R_l(m_l)$, is simply $R_l(m_l) = c_l(m_l)/\sigma_I^2$. This indicates which input distribution contributes the most to IU and therefore where further input data collection may be required.

3.4 Empirical Evaluation

In this section we empirically evaluate and compare our methods using a tractable $M(t)/M/\infty$ queueing model. An illustration of using the methods to quantify IU in a realistic call center setting is also presented to highlight the need for IU quantification in simulation models with nonhomogeneous input processes.

3.4.1 $M(t)/M/\infty$ Queueing Model

We firstly evaluate our methods by considering the $M(t)/M/\infty$ queueing model since it has well-known behaviour and calculation of the contribution of the i^{th} input distribution, $\mathrm{Var}[\mathrm{E}(\bar{Y}(\widehat{F}_i))|\widehat{F}_i]$ for $i = 1, 2, \ldots, p$, is analytically possible. We can therefore assess the quality of the proposed methods from §3.3 against the true values and compare their respective performance.

We investigated the effect of the size of the observed samples of arrival counts and the speed of convergence to steady state within each interval on the performance of our methods. Notice that fast convergence to steady state is analogous to having q distinct $M/M/\infty$ queues and for stationary input distributions we know the mean-variance (M-V) and Taylor series approximation (TSA) methods both perform well. The system performance measure we selected was the expected number in the system over the whole period, $\mathrm{E}(\bar{N})$, which for an infinite server system is also the expected number of busy servers. This measure is linear in λ_i for $i = 1,2,\ldots,q$ and we therefore expected the approximations to be good.

The experiment is as follows. We considered an $M(t)/M/\infty$ queueing system over a $T = 4$ hour period. The arrival rate was assumed to change hourly according to a piecewise-constant function with rates $\lambda(t) = (\lambda_1,\lambda_2,\lambda_3,\lambda_4)$; the service distribution was assumed to be stationary with service rate ψ. To mimic the effect of input uncertainty, the system was "observed" for m_λ days, recording the arrival counts in each interval, and approximately $m_{\boldsymbol{\theta}} = m_\lambda \times 60 \times (\lambda_1+\lambda_2+\lambda_3+\lambda_4)$ service times were observed, one service time for each arrival. These provided the data for the fitted input models.

The experiment was split into two sub-experiments with different arrival processes and service rates reflecting "quick" and "slow" speeds of convergence to steady state. Within each sub-experiment we tested different values of m_λ to see if the number of observations of the arrival counts has an effect on the performance of either method. To enable comparability between the two sets of experiments, m_λ and $m_{\boldsymbol{\theta}}$ are chosen such that the total number of arrivals is the same for each level of sample size. The square root of the true analytical contribution from each parameter was recorded, for compatibility with the performance measure estimate, along with the percentage relative error of both methods in each scenario. In the M-V method $B = 40$ bootstrap samples each of $r = 500$ replications of the simulation were run. The entire experiment was repeated for $h = 1000$ macro-replications. The averaged results can be found in Tables 3.4.1 and 3.4.2.

When calculating the analytical values there is no formula for calculating $\mathrm{Var}[\mathrm{E}(\bar{Y}(\widehat{F}_\psi))|\widehat{F}_\psi]$, although for large enough ψ a very close approximation exists. This approximation was used in Experiment 1 but for Experiment 2, where $\psi = 0.05$, it leads to over-estimation. We therefore simulated 1000 values of $\widehat{\psi}$ and calculated $\mathrm{E}(\bar{N})$ using the parameter space $(\lambda_1,\lambda_2,\lambda_3,\lambda_4,\widehat{\psi})$. The "analytical" values for $\mathrm{Var}[\mathrm{E}(\bar{Y}(\widehat{F}_\psi))|\widehat{F}_\psi]$ reported in Table 3.4.2 are therefore the standard deviation of the 1000 observations of $\mathrm{E}(\bar{N})$.

Notice that the analytically calculated contributions for λ_4 and ψ are smaller in Experiment 2 compared to Experiment 1. When convergence to steady state is slow more work is carried out outside of our window of observation and therefore more service times are truncated by the end of the time period. This causes a reduction in variance which explains the discrepancy between the contributions for λ_4 and ψ across the two experiments. All other values match very closely between the experiments because virtually all the work originating in the first three intervals is completed by the end time, 240 minutes, even in the system that settles to steady state more slowly.

From Tables 3.4.1 and 3.4.2 it is clear that as the amount of input data increases the contributions decrease, as they should. However, our interest here is in the relative errors of contribution estimation for the M-V and TSA methods. When the contributions are small, precise estimation of them is harder. However, the TSA method is asymptotically valid as the number of observations tends to infinity so it tends to hold its relative error level across sample sizes. The M-V approach, on the other hand, has relative errors smaller than

Table 3.4.1: Experiment $1(i)$: The analytical contribution of the i^{th} input distribution and the percentage relative errors of the M-V and TSA methods when the arrival process is $\lambda(t) = \left(\frac{1}{3}, \frac{1}{2}, \frac{5}{12}, \frac{1}{3}\right)$ and service rate $\psi = 0.2$. Here $\mathrm{E}(\bar{N}) = 1.94$.

		$\sqrt{\mathrm{Var}[\mathrm{E}(\bar{Y}(\widehat{F_i}))\|\widehat{F_i}]}$						
Sample Size	Method	λ_1	λ_2	λ_3	λ_4	ψ	Total	Magnitude
$m_\lambda = 2$	Analytical	6.59	8.07	7.37	6.04	14.4	20.1	$\times 10^{-2}$
$m_{\boldsymbol{\theta}} = 190$	M-V (RE%)	0.41	0.30	-0.12	0.22	-3.22	-1.53	
	TSA (RE%)	0.22	0.79	0.62	0.46	-5.93	-2.69	
$m_\lambda = 20$	Analytical	2.08	2.55	2.33	1.91	4.54	6.37	$\times 10^{-2}$
$m_{\boldsymbol{\theta}} = 1900$	M-V (RE%)	0.79	0.29	0.28	0.67	-3.31	-1.44	
	TSA (RE%)	-0.09	2.52	-0.18	0.53	-3.73	-1.45	
$m_\lambda = 100$	Analytical	0.93	1.14	1.04	0.85	2.03	2.85	$\times 10^{-2}$
$m_{\boldsymbol{\theta}} = 9500$	M-V (RE%)	3.65	2.06	1.91	3.17	-1.78	0.38	
	TSA (RE%)	1.67	3.25	0.56	1.32	-3.54	-0.86	

Table 3.4.2: Experiment $1(ii)$: The analytical contribution of the i^{th} input distribution and the percentage relative errors of the M-V and TSA methods when the arrival process is $\lambda(t) = \left(\frac{1}{12}, \frac{1}{8}, \frac{5}{48}, \frac{1}{12}\right)$ and service rate is $\psi = 0.05$. Here $\mathrm{E}(\bar{N}) = 1.84$.

		$\sqrt{\mathrm{Var}[\mathrm{E}(\bar{Y}(\widehat{F}_i))\vert\widehat{F}_i]}$						
Sample Size	Method	λ_1	λ_2	λ_3	λ_4	ψ	Total	Magnitude
$m_\lambda = 8$	Analytical	6.59	8.06	7.25	4.50	12.6	18.4	$\times 10^{-2}$
$m_{\boldsymbol{\theta}} = 190$	M-V (RE%)	-0.46	-0.34	-0.03	-0.11	-2.87	-1.20	
	TSA (RE%)	-1.02	-0.57	-0.69	-0.16	-5.76	-2.28	
$m_\lambda = 80$	Analytical	2.08	2.55	2.29	1.42	4.00	5.84	$\times 10^{-2}$
$m_{\boldsymbol{\theta}} = 1900$	M-V (RE%)	-2.39	-1.27	-2.06	-4.19	-2.27	0.07	
	TSA (RE%)	-1.27	-2.53	-0.93	-1.87	-3.62	-0.77	
$m_\lambda = 400$	Analytical	0.93	1.14	1.03	0.64	1.80	2.62	$\times 10^{-2}$
$m_{\boldsymbol{\theta}} = 9500$	M-V (RE%)	-5.50	-3.96	-4.67	-10.02	-0.3	2.66	
	TSA (RE%)	-1.49	-3.24	-2.79	-2.64	-1.41	0.74	

TSA when m_λ is small, but as m_λ increases the approximate nature of the mean-variance effects model causes the relative errors to increase somewhat. Overall, the M-V method seems to be better when m_λ is small, and TSA is better as m_λ becomes larger. The speed of convergence of the queue to steady state does not seem to affect the performance of our methods for our chosen performance measure. Overall both methods can be said to perform well with most approximations having relative error less than 5%.

3.4.2 Healthcare Call Centre

We will now illustrate the impact of IU quantification in the simulation of a real-world system with a nonhomogeneous input process. We have data from an NHS 111 healthcare call centre. In the UK these call centres are used to advise people who have symptoms of an illness but are unsure where to get treatment. The aim is to reduce congestion in hospital

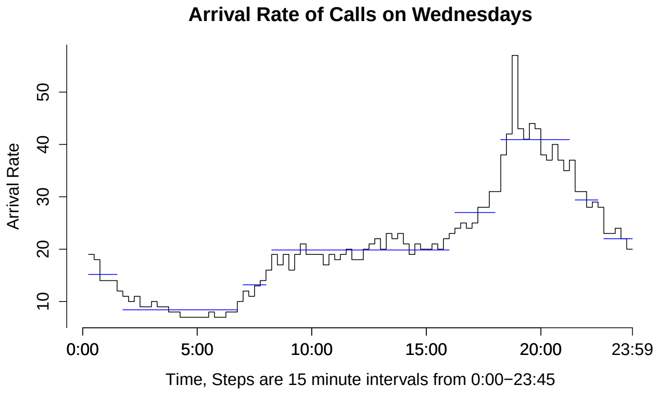


Figure 3.4.1: Change point analysis on the arrival rate of calls on Wednesdays.

EDs or doctors surgeries caused by minor complaints.

The data was split into 96, 15-minute intervals spanning 24 hours. Of the 6 months of data we decided to consider Wednesdays only as public holidays are unlikely to fall mid-week and therefore we would expect no spikes in the arrival rate. Having 6 months worth of data meant we had $m_\lambda = 26$ Wednesdays to consider and these were averaged to find the mean arrival rate within each interval which became our initial piecewise-constant arrival rate function. While it is clear from the mean arrivals in each time interval that the process is not stationary, extended periods of time where the rate was approximately constant could also be observed. Further, it will be difficult to estimate, and not very meaningful to measure, contributions from 96 tiny intervals. Therefore, rather using a large number of small intervals, or choosing arbitrary large intervals, change-point analysis from Chen and Gupta (2011) was applied to let the data guide when to merge periods where the arrival rates were not significantly different. This resulted in 8 periods of differing length as seen in Figure 3.4.1. We would argue that this approach should be used routinely. For the purposes of this analysis, we assume there is no uncertainty in the location of these change points.

In a realistic call centre not only does the arrival rate change with time but so too does the number of servers. Therefore, we simulated the 111 call centre as an $M(t)/G/s(t)$ queue. From two months of service time-data the mean service time was 8.00 minutes and the standard deviation was 4.33 minutes. A moment matching approach was used to fit a Gamma distribution with shape parameter $\psi_1 = 3.408$ and scale parameter $\psi_2 = 2.347$. For the simulation itself, since we wanted to mimic having observed a service time for each arrival, we generated a synthetic "observed" service-data set from the fitted service distribution. The synthetic data set was of size $m_{\boldsymbol{\theta}} = 52,711$ corresponding to the expected number of arrivals, and was treated as the real-world data during the simulation.

The call centre's target level of service is $\mathrm{P}(\mathrm{Wait} > 1\,\mathrm{min}) \leq 0.05$ for each caller. Approximately proportional staffing was applied to each time interval and it was found that the waiting time target was met at a level equivalent to 60% utilisation. This is our base case in the experiment as it is likely to be close to the true staffing level the call centre used. We also simulated the system with constant staff size tuned to the expected arrival rate over the whole day (Case 1) and to 1.5 times this expected arrival rate (Case 2). These staffing patterns are chosen as they highlight the danger of using stationary approximations of input distributions. In practice someone may use the expected arrival rate over the whole day to set a staffing schedule, ignoring the possibility of fluctuation in the arrival rate.

We investigated performance measures such as the probability of waiting more than 1 minute to be served $\mathrm{P}(\mathrm{Wait} > 1\,\mathrm{min})$, the expected number of people in the queue, $\mathrm{E}(\bar{N})$, and the expected waiting time of customers, $\mathrm{E}(\mathrm{WTime})$ over the whole day. The results for the last of these, $\mathrm{E}(\mathrm{WTime})$, can be seen in Table 3.4.3. We used $B = 40$ bootstrap samples, for which $r = 100$ replications of the simulation were carried out for the M-V method. This process was repeated for $h = 1000$ macro-replications of the entire experiment.

Notice first that M-V and TSA give similar, but not identical results. However, they agree on which intervals are the highest and lowest contributors. In Case 1 the contribution of interval 6, $\mathrm{Var}[\mathrm{E}(\bar{Y}(\widehat{F}_{\lambda_6}))|\widehat{F}_{\lambda_6}]$, is much larger than the contribution of any of the other intervals. This coincides with the spike in arrival rate in Figure 3.4.1. At this point the queue would be experiencing very high levels of congestion, the number of servers equates to a utilisation of 112.3% which means all servers are always busy. This also seems to have a knock on effect into the next interval, where the contribution of λ_7 is much higher than the contribution of λ_5 even though they have a similar arrival rate. This may be explained by

Table 3.4.3: The effect of different staffing schemes on the parameter contribution for input distributions, $\mathrm{Var}[\mathrm{E}(\bar{Y}(\widehat{F_i}))|\widehat{F_i}]$, $i = 1,2,\ldots,p$.

		$\mathrm{Var}[\mathrm{E}(\bar{Y}(\widehat{F_i}))\|\widehat{F_i}]$										
Case		λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	ψ	Mag	E(WTime)
Base	M-V	1.27	9.47	1.31	2.31	0.53	0.51	0.60	0.66	2.41	$\times 10^{-6}$	0.0674
	TSA	1.04	9.37	1.12	2.10	0.34	0.32	0.41	0.45	2.45	$\times 10^{-6}$	0.0674
1	M-V	2.11	2.24	2.22	2.34	2.41	165.53	8.03	2.17	17.55	$\times 10^{-3}$	5.17
	TSA	0.51	0.54	0.57	0.64	0.55	162.16	6.26	0.56	15.57	$\times 10^{-3}$	5.17
2	M-V	3.29	3.27	3.31	3.36	3.28	86.64	3.12	3.15	11.32	$\times 10^{-7}$	0.026
	TSA	2.19	1.99	1.91	2.11	2.09	85.45	1.87	2.09	10.88	$\times 10^{-7}$	0.026

both the backlog of customers and the arrival rate being above average in the 7^{th} interval. Although the queue is trying to empty, congestion is still high leading to higher uncertainty. By the final interval the system has recovered from the high congestion levels and the contribution of λ_8 is relatively small.

We see a similar but less pronounced effect in Case 2 where again the contribution of λ_6 is larger than the others. This illustrates the importance of understanding the dynamics of IU as the results show how sensitive the overall estimate of performance is to the correct value of arrival rate during the short 6^{th} interval. In the base case we do not see these patterns, the arrival distribution contributions appear to be similar in all but the second interval. When considering E(WTime) the second interval is the most influential; due to the low number of servers this higher contribution was therefore expected.

3.5 Conclusion

This chapter presents two methods for quantifying input uncertainty in simulation models with NHPP input processes. The key is the use of count observations to model the arrival processes, meaning each interval of the piecewise-constant rate function can be treated as a distinct, stationary input distribution. From this it is simple to calculate the total contribution to IU of each process and therefore the overall IU. Exploiting the fact that the arrivals are Poisson also allowed us to greatly streamline the method based on a mean-variance effects model.

An evaluation of the performance of the methods was presented using the tractable $M(t)/M/\infty$ model; both methods were seen to perform well. An illustration of a realistic call centre scenario was also used to show how input uncertainty quantification in arrival processes may be applied in practice, including the use of change-point analysis to allow the arrival data to guide the choice of time-interval sizes. An open question remains as to the IU that arises from the location of these change points and whether this should be taken into account within the analysis.

Detecting Bias due to Input Modelling in Computer Simulation

4.1 Introduction

In stochastic simulation the "stochastic" element of the simulation comes from the input models that drive it. In this chapter we focus on parametric input models, probability distributions or stochastic processes, that are estimated from observations of the real-world system of interest. Since we can only ever collect a finite number of observations, error, with respect to what the simulation says about the real-world system performance, is inevitable.

Error caused by input modelling can be broken down as MSE = Variance + Bias^2; that is, the mean squared error (MSE) due to input modelling is made up of the variability of the simulation response caused by input modelling, known in the literature as input uncertainty (IU=Variance), and bias due to input modelling squared. Barton (2012) explains that, even in very reasonable simulation scenarios, analysis of the response of interest can be very different when error due to input modelling is included. Barton (2012) was referring to IU, but the same idea holds for bias due to input modelling. In simulation models where a large number of replications of the simulation are completed, effectively driving out the inherent simulation noise caused by random-variate generation, ignoring input modelling uncertainty can lead to over-confidence in the simulation results. Underestimating the error of the simulation response is dangerous, especially when this output may be used to guide important decisions about the real-world system.

To date the main focus of research in this area has been on input uncertainty quantification. Bias due to input modelling has been virtually ignored. This was partially justified by the knowledge that as the number of real-world observations of the input models increases, the bias due to input modelling decreases faster than input uncertainty: given m observations from an input model, it is known that IU is $O(1/m)$, whereas bias squared due to input modelling is typically $O(1/m^2)$ (Nelson, 2013). Despite this, bias can still be substantial for finite m, and should not be ignored.

To facilitate understanding, we consider the simulation of a healthcare call centre. More specifically, we look at the UK National Health Service (NHS) 111 system. NHS 111 was designed to take some of the strain from healthcare systems in the UK, for example, emergency departments and doctors' surgeries. Ringing NHS 111 allows a caller to talk to a healthcare professional who can advise them on what care they need. The NHS 111 call centre can be represented as a stochastic queueing model driven by a non-stationary arrival process and a stationary service distribution. Since we only have a finite number of observations from which to estimate these input models, they are not correct; this error propagates through the NHS 111 simulation model to the performance measures of interest that might be used for staffing.

This chapter presents a delta method approach to estimating bias caused by input modelling in stochastic simulation models. The delta method is based on a second-order Taylor series approximation and therefore requires the quantification of the second-order partial derivatives of the response surface. In simulation, the response of interest is most often an unknown function of its input models which means we cannot directly evaluate its derivatives. We therefore propose use of an experiment design to fit a response surface model from which the second-order partial derivatives can be estimated.

As a key feature of this chapter, we also present a bias detection test with controlled power for detecting bias due to input modelling greater than a pre-chosen threshold value, γ, considered to be the smallest bias of a relevant size. Note that throughout this chapter when we refer to ‘a bias of a relevant size’ we mean a bias that is of a size that would concern a practitioner. In this way when bias is small, and therefore not of concern to us, we require less computational effort to conclude that bias is not significantly different from zero than to accurately estimate it. Also, when bias is large, i.e., greater than γ, we have a high probability of detecting it. In §4.3.1 we describe the novel way in which we construct the experiment design used to estimate the response surface, which allows a practitioner to easily control the power of the bias detection test.

The bias detection test also hinges on our choice of a “bias of relevant size.” When there is no clear choice for γ from the problem context, we propose using the estimated value of IU as a benchmark value: if bias is small fraction of IU, then it contributes little to the overall MSE, while if it is large fraction of IU then it should not be ignored. In §4.4.4 IU is used to guide the choice of relevant bias, γ, for the NHS 111 system.

We begin this chapter with a discussion of current literature in §4.2. In §4.3 we present our delta method approach to bias estimation and the diagnostic test along with an algorithm to aid implementation. In §4.4 we evaluate the performance of the bias diagnostic test including completing a controlled experiment to evaluate the diagnostic test for response functions with different forms, under varying numbers of observations and replications in §4.4.3. In §4.4.4 a realistic application of the method in the NHS 111 healthcare call centre setting is given. We conclude in §4.5. All proofs are left to the appendix.

4.2 Background

To date estimating IU has been the main focus of research in the area of quantifying error caused by input modelling. See Song et al. (2014) for a careful definition and discussion of IU quantification techniques. A number of methods for quantifying input uncertainty in simulation models exist covering both frequentist and Bayesian methodologies (Barton, 2012). Of these, Cheng and Holland (1997) present a delta-method approach for quantifying IU in simulation models with time homogeneous parametric input distributions; in Chapter 3 we extended this method to piecewise-constant non-stationary Poisson arrival processes. In §4.4 this IU quantification method will be used to estimate IU and thus guide our choice of relevant bias.

When one refers to quantifying the 'bias' it is typically the bias of an estimator of a population parameter given a sample of data, averaged over the distribution of possible samples. In our computer-simulation context this bias is also averaged over the natural noise due to generating samples of the stochastic inputs. Stated differently, our estimator is a function of both real-world and simulated sampling. Standard methods for bias quantification are the jackknife and the bootstrap (Efron, 1982), with the jackknife often considered the go-to choice. However, for bias estimation without simulation noise, Withers and Nadarajah (2014) found both the jackknife and the bootstrap inferior to the delta method in terms of computational efficiency in all but a few special cases where it could be said the jackknife method was comparable. When there *is* simulation noise the number of simulation replications required to mitigate it grows as $O(m^2)$, meaning that the simulation effort could become prohibitive or the estimate of bias could be obscured by the simulation noise when m is large; for a proof of this result see Appendix A.1. For a review of the conditions under which the delta method approximation is accurate see Oehlert (1992).

The delta method requires the second-order partial derivatives of the expected value of the simulation response. Since the expected value of the simulation response is not known, we propose using an experimental design to fit a response surface model of it. To allow estimation of the derivatives of the response surface, we require a Resolution V, or higher, experimental design to ensure no confounding of the second-order interactions (Montgomery, 2013). We therefore propose to use a central composite design (CCD) to fit a quadratic response surface model. The CCD is easy to understand and meets the design resolution requirement, but does suffer in terms of scalability requiring an exponentially increasing number of design points as the number of input parameters increases. Fractional factorial designs are one way of reducing the number of design points required to fit a response surface. However, few efficient generators exist for creating Resolution V fractional factorial designs with a large number of inputs. We use the method of Sanchez and Sanchez (2005) to reduce the number of design points needed to support the quadratic response surface. This method can generate designs with over 120 inputs. Methods for creating Resolution V fractional designs are also discussed by Montgomery (2013) and Box (1978) but the allowable number of inputs within these designs is limited.

Neither quantification nor detection of bias due to input modelling have previously been considered. In the following section we present the methodology behind our delta method estimate of bias and our bias detection test.

4.3 Detecting bias of a relevant size

Let there be L parametric input distributions that drive the simulation with, $k \geq L$, true input parameters, $\boldsymbol{\theta}^c = \{\theta_1^c, \theta_2^c, \dots, \theta_k^c\}$. Within the NHS 111 healthcare call centre system, $\boldsymbol{\theta}^c$ are the unknown parameters describing the true arrival process and service distribution. For any set of parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2, \dots, \theta_k\}$, we denote the output of the j^{th} replication of the simulation as $Y_j(\boldsymbol{\theta}) = \eta(\boldsymbol{\theta}) + \varepsilon_j$, where $\eta(\boldsymbol{\theta})$ is the expected value of the simulation response of interest; this could be, for example, the expected waiting time of callers which is our performance measure of interest in §4.4.4. Here we assume ε_j, for $j = 1, 2, \dots, r$, are i.i.d random variables, with mean zero and variance σ^2, that represent the stochastic estimation error arising within each replication of the simulation, $\varepsilon_j \sim$ i.i.d $(0, \sigma^2)$.

For each of the $l = 1, 2, \dots, L$ input distributions we have m_l real-world observations from which we can find the maximum likelihood estimators (MLEs) of the input parameters, $\boldsymbol{\theta}^{mle} = \{\theta_1^{mle}, \theta_2^{mle} \dots, \theta_k^{mle}\}$. By averaging over r replications of the simulation, driven by $\boldsymbol{\theta}^{mle}$, we gain an estimate of the performance measure of interest. We call this the *nominal experiment*. This experiment can reduce the stochastic estimation error about our response of interest through replication of the simulation but it has no effect on the error due to input modelling which is only affected by $m_1, m_2, \ldots, m_L$.

Bias due to input modelling arises because we only have a finite number of observations of the real-world system from which to estimate $\boldsymbol{\theta}^c$. This type of bias describes how far, on average, our simulation response is from the real-world performance given the error that arises from estimating the input models. Specifically

$$b = \mathrm{E}\left[\eta(\boldsymbol{\theta}^{mle})\right] - \eta(\boldsymbol{\theta}^c) \tag{4.3.1}$$

where the expectation is with respect to the sampling distribution of $\boldsymbol{\theta}^{mle}$. In the NHS 111 system this is the expected value of the difference between the performance of the simulation (e.g., fraction of cases waiting more than 1 minute) and the true underlying performance of the real system. When the simulation response is non-linear in $\boldsymbol{\theta}$, as is usually the case, this bias will always arise; we approximate it using the delta method in an innovative way.

Assuming the expected simulation response, $\eta(\cdot)$, is at least twice continuously differentiable about $\boldsymbol{\theta}^c$ it can be expanded as a Taylor series to second-order

$$\eta(\boldsymbol{\theta}^{mle}) \approx \eta(\boldsymbol{\theta}^c) + d(\boldsymbol{\theta}^{mle})^T \nabla\eta(\boldsymbol{\theta}^c) + \frac{1}{2!} d(\boldsymbol{\theta}^{mle})^T \, \mathrm{H}(\boldsymbol{\theta}^c) d(\boldsymbol{\theta}^{mle}), \tag{4.3.2}$$

where $d(\boldsymbol{\theta}^{mle}) = (\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^c)$ is the difference between the MLEs and the true parameters, $\nabla\eta(\boldsymbol{\theta}^c)$ is the $(k \times 1)$ gradient vector of the response function, and $\mathrm{H}(\boldsymbol{\theta}^c)$ is the $(k \times k)$ Hessian matrix of second-order partial derivatives with respect to the k input parameters which approximates the curvature of the response surface. To ease explanation, let there be m observations collected from each of the L input models. But note that, the following results hold in slightly modified form for $m_1 \neq m_2 \neq \cdots \neq m_L$, provided $m_i / \sum_{j=1}^{L} m_j \to c_i > 0$ for some fixed value c_i as $m \to \infty$. Taking the expectation of (4.3.2), whilst noting that, under mild conditions, $\mathrm{E}\left[d(\boldsymbol{\theta}^{mle})\right] = \mathrm{E}\left[(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^c)\right] \to 0$ as $m \to \infty$, we get the delta approximation of bias,

$$b \approx \frac{1}{2} \, \mathrm{E}\left[d(\boldsymbol{\theta}^{mle})^T \, \mathrm{H}(\boldsymbol{\theta}^c) d(\boldsymbol{\theta}^{mle})\right] = b^{approx},$$

which, after some matrix manipulation, simplifies to

$$b^{approx} = \frac{1}{2}\,\mathrm{tr}(\Omega\,\mathrm{H}(\boldsymbol{\theta}^c)). \tag{4.3.3}$$

Here tr() denotes the trace of a matrix and $\Omega = \mathrm{Var}(\boldsymbol{\theta}^{mle})$ denotes the variance-covariance matrix of the MLEs. For a proof of the of the asymptotic equivalence of b and b^{approx} as $m \to \infty$ see Appendix A.2.

As previously noted $\boldsymbol{\theta}^c$ is unknown; if it were known then there would be no error due to input modelling. In simulation studies it is most often the case that the systems we simulate are complex and no tractable form of our response of interest exists; we will therefore also treat the response function, $\eta(\cdot)$, as unknown. This means the delta approximation of bias, b^{approx}, cannot be evaluated directly; we therefore estimate it by

$$\widehat{b} = \frac{1}{2}\mathrm{tr}(\widehat{\Omega}\,\widehat{\mathrm{H}}(\boldsymbol{\theta}^{mle})). \tag{4.3.4}$$

Evaluation of $\widehat{b}$ requires estimates of both the variance-covariance matrix of the input parameters and the Hessian matrix of second-order partial derivatives. In practice we estimate Ω using $\widehat{\Omega} = \mathrm{I}_0(\boldsymbol{\theta}^{mle})^{-1}/m$ the inverse Fisher information evaluated at $\boldsymbol{\theta}^{mle}$. From this point on, $\widehat{\Omega}$ will refer to this plug-in estimate for $\mathrm{Var}(\boldsymbol{\theta}^{mle})$. This introduces additional error into $\widehat{b}$, but we show this error to be insignificant in §4.4.1 of our experimental evaluation. In the experiment a truly quadratic response was considered, such that $b^{approx} = b$, and the relative error of $\widehat{b}$ to b using the plug-in $\widehat{\Omega}$ was found to be less than 1%.

Estimating the Hessian is more difficult. For this we choose a response surface modelling approach, quantifying the non-linearity of the response surface by investigating the behaviour of $\eta(\cdot)$ close to $\boldsymbol{\theta}^{mle}$, our estimate of $\boldsymbol{\theta}^c$. See §4.3.1 below.

Based on our estimate of bias, we present a bias detection test with high power for detection when $|b| \geq \gamma$. In the following sections we illustrate the use of experimental design for estimating the Hessian, and therefore the bias. We also present a novel way to construct this experimental design that allows a practitioner to control of the power of our bias detection test.

4.3.1 Estimating the Hessian

To estimate the Hessian we make the further assumption that our response surface is locally quadratic; that is, near to $\boldsymbol{\theta}^c$

$$\eta(\boldsymbol{\theta}) = \beta_0 + \boldsymbol{\theta}^T\boldsymbol{\beta} + \frac{1}{2}\boldsymbol{\theta}^T\mathbf{B}\boldsymbol{\theta}, \tag{4.3.5}$$

where $\boldsymbol{\beta}$ is the vector of coefficients belonging to the linear terms, $\mathbf{B}$ is the $(k \times k)$ matrix of coefficients belonging to the interaction and quadratic terms and $\boldsymbol{\theta}$ is some vector of input parameter values. Note that, if $\eta(\cdot)$ is twice continuously differentiable at $\boldsymbol{\theta}^c$, as assumed for result (4.3.2), then this is approximately true using Taylor series. In §4.3.3 we suggest a test for lack-of-fit of the quadratic response surface, but for now we will assume (4.3.5) holds; then in §4.4.3 we evaluate this assumption by considering responses with different functional forms.

By fitting this model we can estimate the Hessian matrix of second-order partial derivatives, allowing the evaluation of $\widehat{b}$. It is clear that taking the second-order partial derivatives of (4.3.5), with respect to $\boldsymbol{\theta}$, equates to estimating $\mathbf{B}$. As $\boldsymbol{\theta}^c$ is unknown, we will use a central composite design (CCD), centred at $\boldsymbol{\theta}^{mle}$, to fit this model. The CCD is well known and has Resolution V, allowing the estimation of quadratic and interaction effects without confounding. Figure 4.3.1 illustrates a CCD design in $k = 2$ dimensions; factorial (purple) and axial (yellow) design points are positioned relative to $\boldsymbol{\theta}^{mle}$, the central (red) design point.

To fit model (4.3.5), we complete r replications of the simulation model at each design point. Let n_F denote the number of factorial design points and n_A the number of axial design points. As suggested by Montgomery (2013), we will carry out more replications of the experiment at the centre point allowing more information collection at $\boldsymbol{\theta}^{mle}$, the point at which we wish to estimate the Hessian. We let this number be a multiple of r, which allows us to treat the multiple replications at centre point as $n_C > 1$ design points. The total number of design points n is therefore $n = n_F + n_A + n_C = 2^k + 2k + n_C$, which depends on the number of input parameters, k. The total number of replications is $n \times r$.

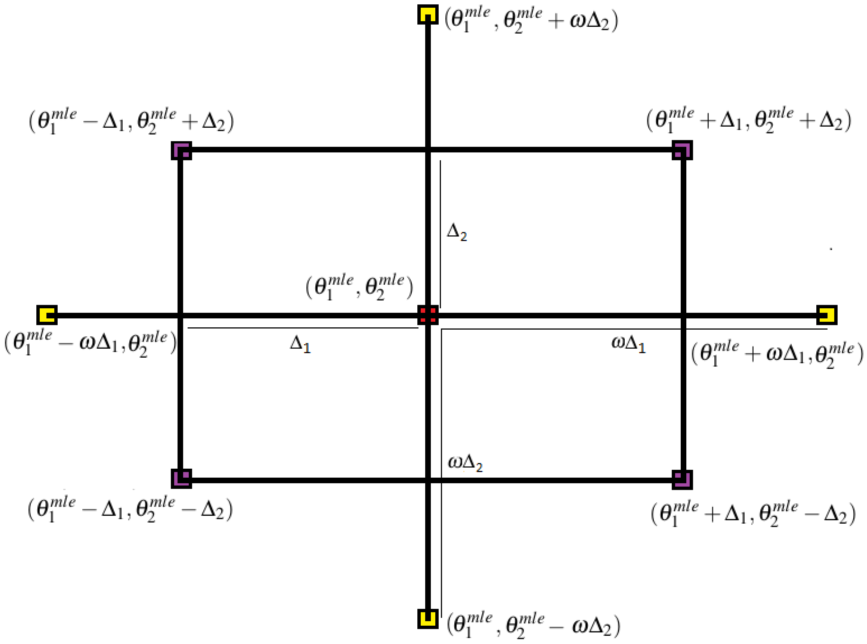

$(\theta_1^{mle}, \theta_2^{mle} + \omega\Delta_2)$
$(\theta_1^{mle} - \Delta_1, \theta_2^{mle} + \Delta_2)$
$(\theta_1^{mle} + \Delta_1, \theta_2^{mle} + \Delta_2)$
Δ_2
$(\theta_1^{mle}, \theta_2^{mle})$
$(\theta_1^{mle} - \omega\Delta_1, \theta_2^{mle})$
Δ_1
$\omega\Delta_1$
$(\theta_1^{mle} + \omega\Delta_1, \theta_2^{mle})$
$\omega\Delta_2$
$(\theta_1^{mle} - \Delta_1, \theta_2^{mle} - \Delta_2)$
$(\theta_1^{mle} + \Delta_1, \theta_2^{mle} - \Delta_2)$
$(\theta_1^{mle}, \theta_2^{mle} - \omega\Delta_2)$

Figure 4.3.1: A CCD design with dimension $k = 2$.

Clearly, the total number of design points, n, grows exponentially with the number of input parameters, k. For $k = 10$, the number of factorial design points is $n_F = 2^{10} = 1024$, even without considering the axial and centre points of the design. We therefore propose the use of fractional factorial designs, with the addition of axial and centre points, to reduce the size of the design. The key to this is to select a Resolution V, or higher, fractional factorial design to ensure no main effects or two-factor interactions are confounded (Montgomery, 2013).

Sanchez and Sanchez (2005) provide an efficient algorithm for generating Resolution V CCDs with a greatly reduced number of design points using discrete-valued Hadamard-Walsh functions to describe and generate the design. Their method focuses on specifying highly-fractionated Resolution V fractional factorial designs. After the fractional-factorial design has been generated the centre and axial points can then be added just as in the full CCD. When $k = 10$, Sanchez and Sanchez (2005) recommend $n_F = 128$ factorial design points, resulting in $n = 148 + n_C$ design points in total without specifying n_C. This is computationally much cheaper than the $n_F = 1024$ factorial design points, in total $n =$

$1044 + n_C$ points, needed in the full CCD experiment. In §4.4.4 we implement these reduced designs alongside the full-factorial CCDs in the NHS 111 healthcare call centre setting.

In Figure 4.3.1, we position the factorial and axial points relative to the centre point, $\boldsymbol{\theta}^{mle}$. Let Δ_i be the distance to a factorial point from the centre point in the i^{th} direction, $i = 1, 2, \ldots, k$, and similarly let τ_i be the distance to the axial points. Experimental designs are often used to investigate the operational range of systems. It is therefore common to work with standardised variables, transforming the original quantitative factors to the values +1 and -1, representing the high and low levels of each factor at the edge of the operational space. We use experimental design quite differently. We are not interested in looking at the behaviour of $\eta(\cdot)$ over the entire range of each input variable. Instead, we are interested in assessing the Hessian of the response surface at the unknown $\boldsymbol{\theta}^c$. By using the standard deviation of the MLEs, $\sqrt{\mathrm{Var}(\theta_i^{mle})}$ for $i = 1, 2, \ldots, k$, to scale the experimental design in each direction, we have a reasonable chance of covering $\boldsymbol{\theta}^c$ without having to stretch our design points so wide that we risk violating the quadratic assumption over our design space. Note that, giving similar reasoning we might have chosen to use the variance-covariance matrix of the MLEs, $\mathrm{Var}(\boldsymbol{\theta}^{mle})$, to scale the design. This would take into account dependencies among the input paramters, but would have introduced substantial additional complexity to the method. Given that we cannot prove that either method leads to the optimal design scaling we opt for the simpler option. That is, we set $\Delta_i = a\sqrt{\mathrm{Var}(\theta_i^{mle})}$ and $\tau_i = \omega\Delta_i = a\omega\sqrt{\mathrm{Var}(\theta_i^{mle})}$ where a is the number of standard deviations the factorial points are from the centre point in the i^{th} direction. Here ω is the scaled distance from the centre to the axial points; we set $\omega = \sqrt{(\sqrt{n_F n} - n_F)/2}$ as suggested by Dean and Voss (1999) for creating orthogonal designs, although we note here that due to the assumed quadratic nature of the response surface, orthogonality does not hold.

At the i^{th} design point we run r replications of the simulation returning the averaged output of the simulation, $\bar{Y}(\boldsymbol{\theta}_i)$ for $i = 1, 2, \ldots, n$. Given these outputs we use least-squares regression to fit the response surface model and therefore evaluate the Hessian,

$$\hat{\mathrm{H}}(\boldsymbol{\theta}^{mle}) = \begin{bmatrix} 2\widehat{\mathrm{B}}_{11} & \widehat{\mathrm{B}}_{12} & \ldots & \widehat{\mathrm{B}}_{1k} \\ \widehat{\mathrm{B}}_{21} & 2\widehat{\mathrm{B}}_{22} & & \\ \vdots & & \ddots & \\ \widehat{\mathrm{B}}_{k1} & & & 2\widehat{\mathrm{B}}_{kk} \end{bmatrix}.$$

Given $\widehat{\Omega}$ and $\hat{\mathrm{H}}(\boldsymbol{\theta}^{mle})$ we now can estimate the bias, using $\widehat{b}$, as in Equation (4.3.4).

We can also estimate $\mathrm{Var}(\widehat{b})$. Conditional on the value of $\widehat{\Omega}$, the plug-in estimate of $\mathrm{Var}\,(\boldsymbol{\theta}^{mle})$; $\mathrm{Var}(\widehat{b})$ below accounts for the variability of the Hessian,

$$\begin{aligned}\mathrm{Var}(\widehat{b}) =& \mathrm{Var}\left[\frac{1}{2}\mathrm{tr}(\widehat{\Omega}\widehat{\mathrm{H}}(\boldsymbol{\theta}^{mle}))\right] \\ =& \frac{1}{4}\mathrm{Var}\left[2\sum_{i=1}^{k}\widehat{\mathrm{B}}_{ii}\widehat{\Omega}_{ii} + \sum_{j=1}^{k}\sum_{i=1, i\neq j}^{k}\widehat{\mathrm{B}}_{ij}\widehat{\Omega}_{ij}\right] \\ =& \sum_{i=1}^{k}\sum_{i\leq j}^{k}\mathrm{Var}(\widehat{\mathrm{B}}_{ij})\widehat{\Omega}_{ij}^{2} + 2\sum_{i\leq j}^{k}\sum_{p\leq q,\, ij<pq}^{k}\mathrm{Cov}(\widehat{\mathrm{B}}_{ij},\widehat{\mathrm{B}}_{pq})\widehat{\Omega}_{ij}\widehat{\Omega}_{pq}.\end{aligned}$$

This requires the calculation of $\mathrm{Var}(\widehat{\mathbf{B}})$, the variance-covariance matrix of regression coefficients belonging to the interaction and quadratic terms. Given we estimated $\widehat{\mathbf{B}}$ by least-squares regression $\mathrm{Var}(\widehat{\mathbf{B}})$ is easily obtained using standard regression analysis. In fact, we derived that $\mathrm{Var}\,(\widehat{\mathbf{B}})$ has special form

$$\mathrm{Var}(\widehat{\mathrm{B}}_{ii}) = \frac{\sigma^2 s}{ra^4\widehat{\Omega}_{ii}^{2}}, \qquad \mathrm{Var}(\widehat{\mathrm{B}}_{ij}) = \frac{\sigma^2 f}{ra^4\widehat{\Omega}_{ii}\widehat{\Omega}_{jj}} \quad \text{and} \quad \mathrm{Cov}(\widehat{\mathrm{B}}_{ii},\widehat{\mathrm{B}}_{jj}) = \frac{\sigma^2 g}{ra^4\widehat{\Omega}_{ii}\widehat{\Omega}_{jj}},$$

where, s, f and g are constants independent of the scaling factor a and $\widehat{\Omega}$. We shall exploit the common ra^4 scaling in §4.3.2 when it comes to setting the width of the CCD in our hypothesis test.

Application of our method will always follow a nominal experiment run at $\boldsymbol{\theta}^{mle}$; therefore, we have a natural estimator $\widehat{\sigma}^2$ of the simulation noise σ^2 from that experiment. In practice we will use this as a plug-in estimator in the expressions for $\mathrm{Var}(\widehat{\mathrm{B}}_{ii}), \mathrm{Var}(\widehat{\mathrm{B}}_{ij})$ and $\mathrm{Cov}(\widehat{\mathrm{B}}_{ii},\widehat{\mathrm{B}}_{jj})$.

We derived that, when using a CCD, $\text{Cov}(\widehat{\text{B}}_{ij}, \widehat{\text{B}}_{lm}) = 0$ when $i \neq j$ or $l \neq m$, and therefore, after some simplification, our estimate of $\text{Var}(\widehat{b})$ has the form

$$\widehat{\text{Var}}(\widehat{b}) = \frac{\widehat{\sigma}^2}{ra^4} \left[sk + f \sum_{i=1}^{k} \sum_{j>i}^{k} \frac{\widehat{\Omega}_{ij}^2}{\widehat{\Omega}_{ii}\widehat{\Omega}_{jj}} + gk(k-1) \right]. \tag{4.3.6}$$

Accounting for the variability of $\widehat{\sigma}^2$ within $\text{Var}(\widehat{b})$ would have made little difference to (4.3.6): the $\text{Var}(\widehat{b})$ would be multiplied by a factor of $(nr-k)/nr$, reflecting the degrees of freedom in the estimate of $\widehat{\sigma}^2$, which we would expect to be very close to one since the total number of design points, $n \times r$, is usually much larger than the number of parameters to be estimated, k.

At this point we have presented a method for estimating the bias of the simulation response caused by input modelling and have also provided a variance estimate associated with it. However, in some cases bias will be small and therefore hard to accurately estimate. When bias is small, we are not interested in getting an accurate estimate of $\widehat{b}$. A bias detection test could therefore save us computational effort since we do not require as much accuracy to be able to assess whether to reject a hypothesis as we perhaps would want if we were to use $\widehat{b}$ within the summary of the error about our performance measure. Let γ denote the size of bias due to input modelling that would concern us. We will now present our key idea, a diagnostic test for detecting bias, with controlled power of rejecting the null when $|b| \geq \gamma$.

4.3.2 A bias detection test

We begin by considering the following hypothesis test

$$\text{H}_0 : b = 0 \qquad \text{vs.} \qquad \text{H}_1 : b \neq 0$$

with test statistic $\text{T} = \widehat{b}/\sqrt{\text{Var}(\hat{b})}$. Let the size of the test be denoted by α_1 and the power by $1-\alpha_2$. We shall assume that

$$\frac{\hat{b} - b}{\sqrt{\text{Var}(\hat{b})}} \sim \text{N}(0,1) = \text{Z}, \tag{4.3.7}$$

which is a reasonable approximation since $\widehat{b}$ is a linear combination of asymptotically normally distributed least-squares regression estimators. The key to this test is in controlling the power at a pre-specified level $1-\alpha_2$ so that, when the absolute bias is truly greater than or equal to γ, we have a high probability of rejecting the null hypothesis. We therefore require an experimental design where the following significance and power constraints hold given γ,

$$\mathrm{P}[\mathrm{T} < \mathrm{Z}_{\alpha_1/2}, \mathrm{T} > \mathrm{Z}_{1-\alpha_1/2} \mid b = 0\,] = \alpha_1 \tag{4.3.8}$$

$$\mathrm{P}[\mathrm{T} < \mathrm{Z}_{\alpha_1/2}, \mathrm{T} > \mathrm{Z}_{1-\alpha_1/2} \mid\; |b| \geq \gamma\,] \geq 1-\alpha_2. \tag{4.3.9}$$

Let the true IU of the response of interest be denoted $\kappa = \mathrm{Var}\left(\eta\left(\boldsymbol{\theta}^{mle}\right)\right)$. Using IU quantification techniques we can estimate κ by $\widehat{\kappa}$. We propose that, when the practitioner does not have an obvious value in mind for γ, $\widehat{\kappa}$ can be used to guide this choice. This is a natural suggestion as it looks at bias within the context of the total MSE due to input modelling. If bias is very small compared to $\widehat{\kappa}$ it may not be worth taking into account. Whereas if bias is large compared to $\widehat{\kappa}$ it would be important, and using $\widehat{\kappa}$ to guide our choice of γ will give us high power of rejecting the null.

We know that Equation (4.3.8) is guaranteed by (4.3.7). Constraint (4.3.9) holds when

$$\sqrt{\mathrm{Var}(\hat{b})} \leq \frac{\gamma}{Z_{1-\alpha_2} - Z_{\alpha_1/2}}. \tag{4.3.10}$$

This says that the variance of our bias estimator, $\mathrm{Var}(\hat{b})$, can be used to control the power of our test. From Equation (4.3.6) it can be seen that, of the components that make up $\widehat{\mathrm{Var}}(\hat{b})$, only the width of the CCD, controlled via a, and the number of replications at each design point, controlled via r, can be influenced by the practitioner. In many simulation scenarios we are constrained by some fixed simulation budget. When this is the case, and we have a set total budget $n \times r$ that we are willing to spend, we can set a, the scaling parameter of the experimental design, to the smallest value such that

$$a \geq \left[\frac{\widehat{\sigma}^2 t^2}{r\gamma^2}\left(sk + f\sum_{i=1}^{k}\sum_{j>i}^{k}\frac{\widehat{\Omega}_{ij}^2}{\widehat{\Omega}_{ii}\widehat{\Omega}_{jj}} + gk(k-1)\right)\right]^{\frac{1}{4}}, \tag{4.3.11}$$

where $t = Z_{1-\alpha_2} - Z_{\alpha_1/2}$ is the difference of the Z-scores given our size and power requirements. Alternatively, we may wish to choose a just large enough so that we can be confident that $\boldsymbol{\theta}^c$ has been covered within the CCD design space and set r appropriate to it; recall that a was defined in units of the standard deviation of the MLEs. Notice that we can easily rewrite (4.3.11) to yield the number of replications as a function of a. Some caution is advised as $r = O(1/a^4)$, which means that a small decrease in the width of the design leads to a great increase in the number of replications required at each design point to control the power.

Due to the limitations on how far we can spread our design before the quadratic assumption breaks down, we propose fixing an appropriately large r and letting (4.3.11) guide our choice of a. In §4.3.3 we describe a lack-of-fit test that can be used to test the quadratic assumption.

Given a and r that satisfy (4.3.11), we are able to set up the CCD to ensure that power holds at the pre-set level, $1 - \alpha_2$, within the hypothesis test. We can now carry out the bias detection test knowing that if bias is truly greater than or equal to γ we have a high probability of rejecting the null.

On completion of the test, even if we reject H_0, we cannot say anything about the size of the bias. We have sufficient evidence to suggest that the bias is non-zero at the $\alpha_1\%$ level, and therefore is worth considering within the error about our response, but we cannot be sure that it is greater than or equal to our relevant value of bias γ. At this point the practitioner may wish to collect further observations of the real system to reduce error due to input modelling. Another option might be to spend further simulation effort on improving the precision of the estimate $\widehat{b}$ so it can be included in a summary of the total error of the response. Whichever choice is made we have presented a novel method for detecting bias due to input modelling, a source of error that, before this contribution, had been ignored.

An algorithm for the bias diagnostic test is summarised below.

0. Preliminary Step. From the real-world observations estimate $\boldsymbol{\theta}^c$ and Ω by $\boldsymbol{\theta}^{mle}$ and $\widehat{\Omega}$. From the nominal experiment estimate σ^2 by $\widehat{\sigma}^2$. Set γ, a bias we wish to detect,

α_1 the size, and $1-\alpha_2$ the power, of the test.

1. To ensure the power holds: initially let $a = 1$, noting that any positive value will suffice; create the $\left(n \times \left(1+2k+\frac{k(k-1)}{2}\right)\right)$ design matrix X, centred at $(0,0,\ldots,0)$ with $\Delta_i = a\sqrt{\mathrm{Var}(\theta_i^{mle})}$ and $\tau_i = \omega\Delta_i$, for $i = 1,2,\ldots,k$. Given X, evaluate s, f and g as follows

$$s = (X^{\mathrm{T}}X)^{-1}_{[\frac{(k+1)(k+2)}{2},\frac{(k+1)(k+2)}{2}]}\Delta_k^4, \qquad f = (X^{\mathrm{T}}X)^{-1}_{[k+2,k+2]}\Delta_1^2\,\Delta_2^2,$$
$$g = (X^{\mathrm{T}}X)^{-1}_{[\frac{(k+1)(k+2)}{2}-1,\frac{(k+1)(k+2)}{2}]}\Delta_{k-1}^2\,\Delta_k^2$$

   where the subscript $[i,j]$ denotes the element in the ith row and jth column of a matrix. Now use (4.3.11) to set a and r, to ensure power holds.

2. Re-build the design matrix X, centred at $(\theta_1^{mle},\theta_2^{mle},\ldots,\theta_k^{mle})$, given a.

3. For each design point $i = 1,2,\ldots,n$, run r replications of the simulation at $\boldsymbol{\theta}_i$, corresponding to row i of the design matrix; average over the r replications to find $\bar{Y}(\boldsymbol{\theta}_i)$.

4. Using the simulation output from each design point $\bar{Y}(\boldsymbol{\theta}_i)$, for $i = 1,2,\ldots,n$, estimate the regression coefficients $(\widehat{\beta}_0,\widehat{\beta}_1,\widehat{\beta}_2,\ldots,\widehat{\beta}_k,\widehat{\mathrm{B}}_{11},\widehat{\mathrm{B}}_{12},\ldots,\widehat{\mathrm{B}}_{(k-1)k},\widehat{\mathrm{B}}_{kk})^T = (\mathrm{X}^T\mathrm{X})^{-1}\mathrm{X}^T\bar{\boldsymbol{Y}}(\boldsymbol{\theta})$, giving $\widehat{\mathrm{B}}_{11},\widehat{\mathrm{B}}_{12},\ldots,\widehat{\mathrm{B}}_{(k-1)k},\widehat{\mathrm{B}}_{kk}$.

5. Evaluate $\widehat{\mathrm{H}}(\boldsymbol{\theta}^{mle})$; thus, estimate b and $\mathrm{Var}(\widehat{b})$ by $\widehat{b}$ and $\widehat{\mathrm{Var}}(\widehat{b})$.

6. Calculate the test statistic, $\mathrm{T} = \widehat{b}/\sqrt{\widehat{\mathrm{Var}}(\widehat{b})}$. If $|\mathrm{T}| \geq Z_{1-\alpha_1/2}$ reject the null hypothesis.

4.3.3 Validating the bias test

Up to this point we made the assumption that our response surface, $\eta(\cdot)$, is truly quadratic near $\boldsymbol{\theta}^c$. In reality we know this does not hold in all cases. For example in §4.4.2 we explore the detection of bias caused by input modelling in a single-server Markovian queue with capacity, C. For this system the expected number of customers in the system in steady state is not quadratic.

In reality the expected response surface is unlikely to be truly quadratic, but as long as the quadratic assumption holds locally within our CCD, we will get a good approximation of the non-linearity of the response surface at $\boldsymbol{\theta}^{mle}$. We therefore propose using a lack-of-fit test to check the quadratic assumption on the response surface by comparing the fit of the assumed quadratic model to the fit of a saturated model with as many coefficients as design points (Montgomery, 2013). That is, comparing the fit of (4.3.5), with $\frac{1}{2}(k+1)(k+2)$ coefficients, to the saturated model, with $1+2k+2^k$, coefficients. In the $k=2$ dimensional case the saturated model would be,

$$\eta(\boldsymbol{\theta}) = \lambda_0 + \lambda_1 z_1 + \lambda_2 z_2 + \lambda_3 z_3 + \lambda_4 z_4 + \lambda_5 z_5 + \lambda_6 z_6 + \lambda_7 z_7 + \lambda_8 z_8 + \varepsilon \tag{4.3.12}$$

where, to identify from which design point each response was measured, indicator variable z_i is equal to 1 at design point i and 0 otherwise. Note that we can take into account the final design point without assigning it an indicator variable by setting $z_i = 0$ for all i. Fitting (4.3.12) requires r replications at each design point.

Testing for lack of fit by comparing the quadratic model to a saturated model comes with certain advantages. Firstly, we do not have to assume any functional form for our response surface; we could have compared the quadratic model to a cubic model for example but there is no guarantee that the cubic part of the model would be the problem in all cases. Also, the saturated model does not require any additional simulation effort to incorporate within our method; we already carry out the r replications required at each design point to fit it.

Running the lack-of-fit test prior to our bias detection test enables us to examine the quadratic assumption. Of course, a hypothesis is just an assessment of evidence: accepting the null hypothesis does not prove that the approximation of a quadratic surface near $\boldsymbol{\theta}^{mle}$ is good enough to provide a trustworthy estimate of bias. However, rejecting the quadratic fit is a useful warning that the resulting bias estimate might not be trustworthy. By the nature of Taylor series approximation, a smaller-width CCD will tend to imply better conformance to a quadratic approximation. Therefore, one way to react to a significant lack of fit is to increase r, the number of replications at each design point, which leads to shrinking the width-scaling parameter a while preserving the power of the bias test at $1-\alpha_2$ (see §4.3.2 and in particular Equation (4.3.11)). That said, repeated application of the lack-of-fit test with different sample sizes, the unknown effect of the experiment design used to fit the quadratic model, and the power of the lack-of-fit test muddies the overall inference. Thus, while we recommend the lack-of-fit test its conclusions are at best advisory, and standard regression diagnostics applied to the quadratic model will also be helpful.

Running the lack-of-fit test prior to our bias detection test enables us to assess the quadratic assumption. If the test is passed, we can be confident that the quadratic assumption is acceptable. On the other hand, on failing the lack-of-fit test, our estimate of bias, $\widehat{b}$, and thus the conclusion of the bias detection test comes into question. By the nature of Taylor series approximation, the smaller the width of the CCD, the smaller the error in our quadratic assumption. Therefore, one way to remedy the rejection of the quadratic assumption by the lack-of-fit test is to repeat the experiment with increased r, the number of replications at each design point. As discussed in §4.3.2, increasing r in Constraint (4.3.11) shrinks a, the scaling parameter for the width of our CCD, making the quadratic assumption hold more closely whilst holding the power at the pre-specified value $1-\alpha_2$. In the following section, where we empirically evaluate our methods, we incorporate the lack-of-fit test into our bias detection test.

4.4 Empirical Evaluation

In this section we evaluate the diagnostic test presented in §4.3 by considering how well the power holds: firstly in a system where the simulation response surface is truly quadratic, and then for a tractable $M/M/1/C$ queueing model. In §4.4.3 we then complete a controlled study considering four tractable response surfaces with different functional forms whilst controlling the number of input observations, m, and the number of simulation replications at each design point, r. We then demonstrate the use of the bias detection test in the NHS 111 call centre setting in §4.4.4.

4.4.1 A Truly Quadratic Model

Consider a quadratic response function. As an example, when $k = 2$ let the response function be given by

$$\eta(\boldsymbol{\theta}) = 2 + 3\theta_1 + \theta_2 + 4\theta_1\theta_2 + \theta_1^2 + 2\theta_2^2. \tag{4.4.1}$$

Here we let θ_1^c and θ_2^c be the true mean parameters from the following bivariate normal distribution

$$X_1, X_2 \sim \mathcal{N}\left((\theta_1^c, \theta_2^c)^T, \begin{pmatrix} \xi_1^2 & 0 \\ 0 & \xi_2^2 \end{pmatrix}\right)$$

with $\text{Cov}(\theta_1^{mle}, \theta_2^{mle}) = 0$ and $\text{Var}(\theta_i^{mle}) = \xi_i^2/m$. Given this response function we know the Hessian matrix exactly, therefore the delta approximation of bias gives $b^{approx} = \xi_1^2/m + 2\xi_2^2/m$ which is exact, $b^{approx} = b$, since (4.4.1) is quadratic.

Let us now assume that the response function, $\eta(\boldsymbol{\theta})$, is unknown to us. We wish to evaluate the performance of the diagnostic test when the underlying response surface is truly quadratic. To do this we investigate how well the power holds when the relevant bias, γ, is set equal to b^{approx}, the true bias in this quadratic case. For this experiment let the power be set to $1 - \alpha_2 = 0.8$. We therefore wish to illustrate our diagnostic test having probability 0.8 of rejecting the null hypothesis when $\gamma = b^{approx}$.

To show our diagnostic test attains this desired power we run a macro-experiment, repeating the diagnostic test $G = 1000$ times. An estimate of power will be given by the proportion of times the null hypothesis is rejected; we denote this estimate $\widehat{p}$. In Table 4.4.1 $\widehat{p}$ is recorded along with $\bar{\bar{b}}$ and $\widehat{\text{Var}}(\widehat{b})$, the sample mean and variance of the bias estimates recorded over the $G = 1000$ macro-replications. Also reported is b^{approx}, the true bias in this quadratic example, which we set equal to γ, the relevant bias.

To complete the diagnostic test we use the methods presented in §4.3. Given true input parameters $\theta_1^c = 5$ and $\theta_2^c = 2$ with $\xi_1^2 = 2$ and $\xi_2^2 = 1.5$, $m = 40$ observations of X_1 and X_2 were generated from the bivariate normal distribution and used to estimate the MLEs, $\boldsymbol{\theta}^{mle}$, and $\widehat{\Omega}$. We set the number of replications to be run at each design point to $r = 1000$ then

Table 4.4.1: How power holds when $\gamma = b^{approx}$ given a truly quadratic response function.

θ_1^c	θ_2^c	r	m	$b^{approx}\ (=\gamma)$	$\bar{\widehat{b}}$	$\widehat{\mathrm{Var}}(\widehat{b})$	$\widehat{p}$
5	2	1000	40	0.2125	0.2111	4.52×10^{-3}	0.79

built a response surface model using a CCD centred at $\boldsymbol{\theta}^{mle}$ with width $a = 0.283$ selected to ensure a power of $1-\alpha_2 = 0.8$. In each replication we ran the simulation by adding $\mathscr{N}(0, 0.01)$ noise to (4.4.1). Given the response surface model the bias estimator, $\widehat{b}$, and its variance, $\widehat{\mathrm{Var}(\widehat{b})}$, could be evaluated enabling the calculation of the test statistic, T, and the conclusion of the diagnostic test. This process was repeated $G = 1000$ times to gain the results shown in Table 4.4.1.

In Table 4.4.1 we see that when the response function is truly quadratic, the diagnostic test holds power very close to $1-\alpha_2 = 0.8$ as desired. We also see that the average of the bias estimates, $\bar{\widehat{b}}$, is very close to the true bias.

4.4.2 $M/M/1/C$ Queueing Model

Consider an $M/M/1/C$ queueing model with true arrival rate θ_1^c, service rate θ_2^c and finite capacity C. Here inter-arrival times of customers, A_i, follow an exponential distribution $A_i \sim \mathrm{Exp}(\theta_1^c)$, as do the service times, $S_i \sim \mathrm{Exp}(\theta_2^c)$, for $i = 1, 2, \ldots, m$ observations. For this queueing model the expected number of customers in the system, $\mathrm{E}[Y|\boldsymbol{\theta}]$, can be expressed in closed form

$$\eta(\boldsymbol{\theta}) = \mathrm{E}[Y|\boldsymbol{\theta}] = \frac{\theta_1}{\theta_2 - \theta_1} - \frac{(C+1)\theta_1^{C+1}}{\theta_2^{C+1} - \theta_1^{C+1}}. \tag{4.4.2}$$

It is therefore possible to derive the second-order partial derivatives yielding $\mathrm{H}(\boldsymbol{\theta}^c)$; this allows the evaluation of b^{approx}, the delta method approximation of bias.

We shall now, for the purpose of the experiment, assume that the true response function, Equation (4.4.2), is unknown. We want to evaluate the quality of our diagnostic test for detecting a relevant bias when the response function is not truly quadratic. To do this we will look at both the $M/M/1/10$ and $M/M/1/100$ queueing models over a number of parameter settings to see how well the power, set at $1-\alpha_2 = 0.8$, holds when relevant bias, γ, is set equal to the delta approximation of bias b^{approx}. As before, to measure the power we record the proportion of times the null hypothesis was rejected over $G = 1000$ macro-replications of the diagnostic test, $\widehat{p}$. The results of the experiments are given in Table 4.4.2.

The diagnostic test was completed as follows. Instead of running a nominal experiment we used the true input distributions to generate m observations from the arrival and service distributions, A_i, S_i for $i = 1,2,\dots,m$, then estimated the MLEs, $\boldsymbol{\theta}^{mle}$, and the covariance matrix, $\widehat{\Omega}$; we know that $\text{Cov}(\theta_1^{mle}, \theta_2^{mle}) = 0$. Also, rather than directly simulating the $M/M/1/C$ queue we add $\mathcal{N}(0, 0.05)$ noise to (4.4.2) for each replication. The number of replications to be run at each design point was set to $r = 500$ allowing the identification of the value of a required for the power to hold at $1-\alpha_2 = 0.8$. A CCD design, centred at $\boldsymbol{\theta}^{mle}$, was then created using a to set the distance to the design points. Replications of the simulation were run at each design point and the response surface fitted allowing evaluation of $\widehat{\text{H}}(\boldsymbol{\theta}^{mle})$, the estimated Hessian matrix. We were therefore able to estimate the delta approximation of bias, $\widehat{b}$, and its variance, $\text{Var}(\widehat{b})$, allowing us to calculate the test statistic and conclude the hypothesis test. This process was repeated over $G = 1000$ macro-replications giving $\widehat{p}$ and $\bar{\widehat{b}}$, the average of the bias estimates, both are recorded in Table 4.4.2.

In Table 4.4.2, we see that across all experiments, whether $C = 10$ or 100, as the amount of input data is increased $\widehat{p}$ gets closer to the desired power $1-\alpha_2 = 0.8$ and the average bias estimate $\bar{\widehat{b}}$ gets closer to the delta approximation b^{approx}. Both parameter estimates improve due to the the increase in information which sees $\boldsymbol{\theta}^{mle}$ get closer to $\boldsymbol{\theta}^{c}$, the true input parameters. This is important in our method as, ideally, we would centre our CCD at $\boldsymbol{\theta}^{c}$ to find the curvature of the response function at that point, $\text{H}(\boldsymbol{\theta}^{c})$.

Experiments 6, 7 and 8 look at the system under high traffic intensity, $\rho = \theta_1^c/\theta_2^c = 0.833$. In Experiment 6, where $m = 40$, we saw a reasonably high proportion of instances

Table 4.4.2: How power holds when $\gamma = b^{approx}$ given an $M/M/1/C$ queueing model.

			$M/M/1/10$			$M/M/1/100$		
Exp	$\frac{\theta_1^c}{\theta_2^c}$	m	b^{approx}	$\bar{\widehat{b}}$	$\widehat{p}$	b^{approx}	$\bar{\widehat{b}}$	$\widehat{p}$
1	0.25	40	0.019	0.024 (4.98×10^{-4})	0.766	0.019	0.025 (6.26×10^{-4})	0.787
2	0.25	100	0.007	0.008 (1.25×10^{-4})	0.79	0.007	0.009 (1.30×10^{-4})	0.789
3	0.50	40	0.134	0.174 (3.86×10^{-3})	0.704	0.150	0.855 (1.73×10^{-1})	0.659
4	0.50	100	0.053	0.063 (1.07×10^{-3})	0.775	0.060	0.085 (2.98×10^{-3})	0.741
5	0.50	1000	0.005	0.006 (6.77×10^{-5})	0.818	0.006	0.007 (7.80×10^{-5})	0.822
6	0.83	100	0.164	0.114 (3.09×10^{-3})	0.611	3.300	6.623 (1.24×10)	0.611
7	0.83	1000	0.016	0.015 (1.98×10^{-4})	0.712	0.330	0.570 (2.64×10^{-2})	0.713
8	0.83	5000	0.003	0.003 (3.82×10^{-5})	0.777	0.066	0.071 (1.08×10^{-3})	0.765

($\approx$ 10%) where the estimated traffic intensity exceeded 1, i.e. $\rho = \theta_1^{mle}/\theta_2^{mle} > 1$. When this occurs the number of people in the queue will increase up to capacity and remain around that level. The behaviour of the response surface in these cases is not quadratic and therefore the delta method does not perform well which is reflected in the average bias estimate, $\bar{\widehat{b}}$, and power, $\hat{p}$. One way to fix this problem is to collect more data, m, until $\theta_1^{mle}/\theta_2^{mle} < 1$ consistently, as we did in Experiments 7 and 8 where the bias estimate $\bar{\widehat{b}}$ gets closer to the delta approximation.

This problem is not unique to bias estimation: it will occur in any simulation model with finite capacity and traffic intensity close to 1. If the amount of data available is small and we cannot accurately estimate the input parameters it is easy to conclude that a system will become saturated when in reality it might not.

In experiments 6, 7 and 8, where a high traffic intensity was investigated, we see the effect of the shape of the true response surface on how well the power holds. The shape of the response surface is driven by the capacity, C. This directly links to how closely $\boldsymbol{\theta}^c$ can be estimated by $\boldsymbol{\theta}^{mle}$. In Figure 4.4.2 we see that for the $M/M/1/100$ queue, with higher capacity, there is a more dramatic change in the response surface for small changes of θ_1

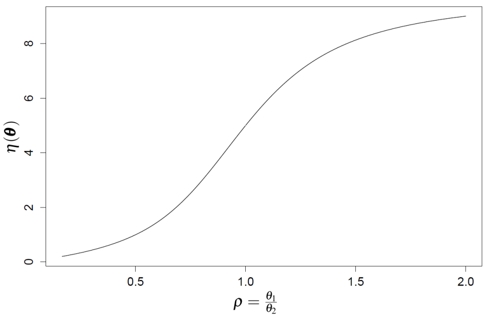


Figure 4.4.1: $M/M/1/10$

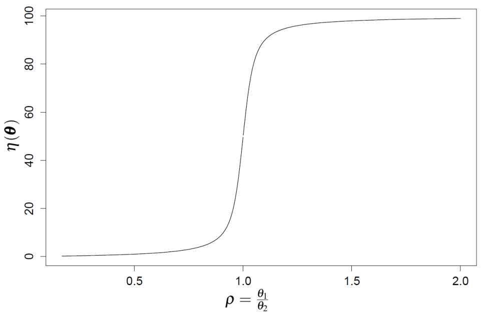


Figure 4.4.2: $M/M/1/100$

and θ_2 than there is for the lower capacity, $M/M/1/10$, queue seen in Figure 4.4.1. Close to $\rho = 1$, where the response surface changes more dramatically, more observations, m, are needed to ensure we are estimating the Hessian, $\mathrm{H}(\boldsymbol{\theta}^c)$, close enough to $\boldsymbol{\theta}^c$ to capture the true curvature at that point. This could also be affected by the variability of the MLEs; when the variance is large even if we have $\boldsymbol{\theta}^{mle}$ close to $\boldsymbol{\theta}^c$ on average, we could see large variability in the response from replication to replication. In the higher capacity system small changes in the inputs have a larger effect on the simulation output which is used to fit the response surface and therefore estimate the Hessian. For the lower capacity queueing model the distance between $\boldsymbol{\theta}^{mle}$ and $\boldsymbol{\theta}^c$ has a less pronounced effect on the simulation response as the response surface changes.

We also note that for the $M/M/1/100$ queueing model, in Experiments 3, 6 and 7 $\widehat{p}$ is lower than the desired power of 0.8 but the average of the bias estimates in these cases, $\bar{\widehat{b}}$, is higher than b^{approx}. Intuitively, this seems contradictory as we would expect to reject the null hypothesis more often if bias is much more extreme than $\gamma = b^{approx}$. In these cases we also see $\bar{\widehat{b}}$ has large standard error. Investigating the test statistics over the $G = 1000$ macro replications, using Q-Q plots, illustrated that these were the cases where the distribution of the test statistics was far from the assumed normal distribution. In Experiments 4, 5 and 8 given more input data the normality assumption was more reasonable. For the $M/M/1/10$ queueing model the normality assumption held well in all cases. This again illustrates the importance of centring the CCD close to $\boldsymbol{\theta}^c$, especially when there is a sharp change in the shape of the response surface.

As an aside we also considered the trade off between the variables a and r, used to set the width of the experimental design. To improve the quadratic assumption it is tempting to shrink a and increase the number of replications at each design point to ensure the power still holds. This is very expensive computationally; to halve a, and thus the width of the design, in the experiments above we would have had to increase the number of replications at each design point to $r = 8000$. Looking at the experiments above we saw little improvement on the estimated power $\widehat{p}$ from halving a. This is because shrinking the width of the design would only be helpful if the CCD was centred very close to $\boldsymbol{\theta}^c$; no amount of computational effort will improve our estimate of $\widehat{\mathrm{H}}(\boldsymbol{\theta}^{mle})$ if the design is centred at $\boldsymbol{\theta}^{mle}$ far from $\boldsymbol{\theta}^c$.

4.4.3 Evaluation of the method given linear, quadratic and cubic response surfaces

Recall that bias due to input modelling is caused when error in the estimation of the input models that drive the simulation is passed through a non-linear response function. We therefore evaluate how well our bias detection test works when there is no bias due to input modelling i.e., the response is linear; when the response surface is truly quadratic; and finally when the underlying quadratic assumption does not hold.

We consider a stochastic simulation model with two unknown input parameters, $\boldsymbol{\theta}^c = \{\theta_1^c, \theta_2^c\} = \{3, 2\}$. These input parameters are the means of two independent exponentially distributed random variables, $W_1 \sim \mathrm{Exp}(1/\theta_1^c)$, $W_2 \sim \mathrm{Exp}(1/\theta_2^c)$.

Within this setting we consider the following functional forms for the response surface $\eta(\boldsymbol{\theta})$: linear, Equation (4.4.3); quadratic, Equation (4.4.4); and two cubic functions, Equations (4.4.5) and (4.4.6), as displayed in Figure 4.4.3,

$$\eta(\boldsymbol{\theta}) = 3 - 10\theta_1 + 4\theta_2 \quad (4.4.3)$$

$$\eta(\boldsymbol{\theta}) = 3 - 10\theta_1 + 4\theta_2 + 8\theta_1\theta_2 + 2.5\theta_1^2 - 2.5\theta_2^2 \quad (4.4.4)$$

$$\eta(\boldsymbol{\theta}) = 3 - 10\theta_1 + 4\theta_2 + 8\theta_1\theta_2 + 2.5\theta_1^2 - 2.5\theta_2^2 + 0.4\theta_1^3 - 0.8\theta_2^3 \quad (4.4.5)$$

$$\eta(\boldsymbol{\theta}) = 3 - 10\theta_1 + 4\theta_2 + 8\theta_1\theta_2 + 2.5\theta_1^2 - 2.5\theta_2^2 + 0.8\theta_1^3 - 3\theta_2^3. \quad (4.4.6)$$

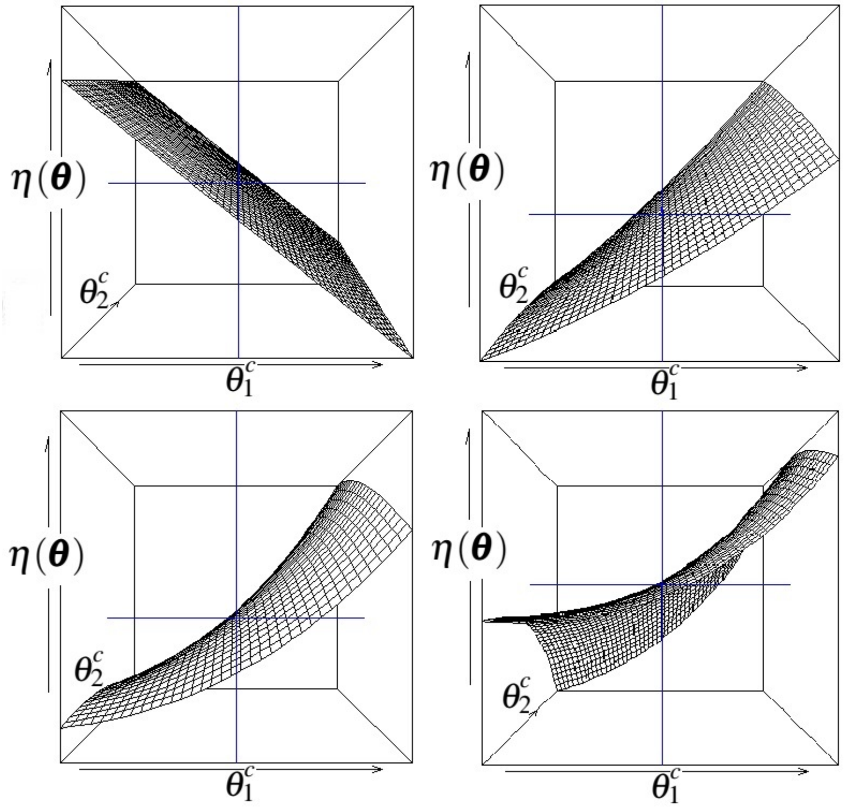


Figure 4.4.3: The true response surfaces plotted over the CCD design space. Top left: linear, Equation (4.4.3); top right: quadratic, Equation (4.4.4); bottom left: cubic, Equation (4.4.5); and bottom right: cubic, Equation (4.4.6). The point(θ_1^c, θ_2^c) is marked in blue.

In this carefully constructed experiment the input parameters and the response functions are known. We also chose our input distributions so that the third moment of the MLE could be calculated exactly and were therefore able to quantify, b, the bias due to input modelling in each system as well as the delta approximation of bias, b^{approx}; see Table 4.4.3. We set the size of the bias detection test to $\alpha_1 = 0.05$ and the power to $1-\alpha_2 = 0.8$; the size for the lack-of-fit test is also 0.05.

To evaluate the bias detection test the value of relevant bias γ is set equal to the delta approximation of bias b^{approx} in both the quadratic and cubic scenarios. In setting $\gamma = b^{approx}$ we expect the power to hold at the pre-set value $1-\alpha_2$. In the linear experiment $b = b^{approx} = 0$, so we use $\widehat{\kappa}$, the estimate of IU, found using the method of Cheng and Holland (1997), to guide the choice of γ where $\gamma = \sqrt{0.3\widehat{\kappa}}$.

Since the true bias, b, is known in these examples we set σ^2/r to be 5 times larger than b in the quadratic and cubic experiments, implying that there is still signficant simulation noise in the evaluation of each design point. In all of the linear experiments σ^2 was set to 0.1. Given σ^2 and the response functions, we simulated by adding normally distributed noise, $N(0,\sigma^2)$, to Equations (4.4.3), (4.4.4), (4.4.5) and (4.4.6). From here on we assume the response functions are unknown and require estimation for the bias detection test.

We complete $G = 1000$ macro-replications of the bias detection test. To do this we collect m observations from each input distribution by generating observations, $\{w_{11}, w_{12}, \ldots, w_{1m}\}$ and $\{w_{21}, w_{22}, \ldots, w_{2m}\}$ from the true input distributions. This is our “real-world” data from which we estimate the input parameters using maximum likelihood. Given these estimates we run the nominal experiment and, in the linear case, estimate the IU in the model. We then apply the bias detection test.

To quantify how well the bias detection test performs we estimate the power of the test by recording the empirical power, the proportion of times we reject the null hypotheses over $G = 1000$ macro-replications; we call this estimate $\widehat{p}$. We then observe how close the empirical estimate $\widehat{p}$ gets to the nominal power, $1-\alpha_2 = 0.8$, for $\gamma = b^{approx}$, given the functional form of $\eta(\cdot)$, m and r. We also record the average of the estimates of bias due to input modelling, $\widehat{b}$, over the G replications, $\bar{\widehat{b}}$, for comparison with the true bias, b. The results are presented in Table 4.4.3.

In the linear system, Equation (4.4.3), there is no bias. In Table 4.4.3 it can be seen that we reject the null hypothesis of no bias and the of lack of fit test in approximately 5% of all

Table 4.4.3: Bias test results varying the form of $\eta(\cdot)$, the amount of input data, m, and number of replications, r. Here $\widehat{p}$ and LOF are the fraction out of $G = 1000$ macroreplications that the bias test and lack-of-fit test, respectively, rejected their null hypothesis, and $\bar{\widehat{b}}$ is the average bias estimate.

	m	r	b	b^{approx}	$\bar{\widehat{b}}$	$\widehat{p}$	LOF
Linear (4.4.3)							
	10	50	0.00	0.00	-0.01	0.06	0.04
	100	50	0.00	0.00	0.00	0.05	0.05
	1000	50	0.00	0.00	0.00	0.04	0.06
	10	500	0.00	0.00	0.00	0.05	0.05
	100	500	0.00	0.00	0.00	0.05	0.05
	1000	500	0.00	0.00	0.00	0.04	0.05
Quadratic (4.4.4)							
	10	50	1.25	1.25	1.36	0.64	0.05
	100	50	0.13	0.13	0.13	0.71	0.06
	1000	50	0.01	0.01	0.01	0.80	0.05
	10	500	1.25	1.25	1.42	0.63	0.06
	100	500	0.13	0.13	0.13	0.72	0.05
	1000	500	0.01	0.01	0.01	0.80	0.06
Cubic 1 (4.4.5)							
	10	50	2.66	2.57	3.01	0.70	0.06
	100	50	0.26	0.26	0.26	0.65	0.06
	1000	50	0.03	0.03	0.03	0.75	0.05
	10	500	2.66	2.57	3.33	0.69	0.06
	100	500	0.23	0.26	0.27	0.70	0.06
	1000	500	0.03	0.03	0.03	0.78	0.06
Cubic 2 (4.4.6)							
	10	50	0.48	0.53	0.08	0.96	0.62
	100	50	0.05	0.05	0.05	0.92	0.22
	1000	50	0.01	0.01	0.01	0.74	0.09
	10	500	0.48	0.53	0.90	0.97	0.36
	100	500	0.05	0.05	0.06	0.92	0.10
	1000	500	0.01	0.01	0.01	0.78	0.07

the linear cases corresponding to the pre-set size of the tests, 0.05, as required.

In the quadratic system, Equation (4.4.4), the delta approximation of bias is exact, so $b^{approx} = b$, and since the response is globally quadratic centring the CCD at $\boldsymbol{\theta}^{mle}$ rather the $\boldsymbol{\theta}^{c}$ does not matter. Therefore, we would expect the power hold at $1-\alpha_2$ plus or minus sampling error. In Table 4.4.3 we see this for $m = 1000$ and it is close for $m = 500$ where the error in $\widehat{p}$ is roughly ± 0.04. When $m = 10$ however, we see a lower power than expected and a discrepancy between $b = b^{approx}$ and $\bar{\bar{\widehat{b}}}$. When the quantity of real-world input data is so exceptionally small, use of the plug-in estimate $\widehat{\Omega}$ without accounting for its variance is likely the reason.

Two cubic functions were also considered. When the response surface is cubic the locally quadratic assumption of our response surface not strictly correct, but it may be reasonable depending on the cubic function. Here b, the true bias due to input modelling, contains the third moment of the MLEs of the input distributions, $\mathrm{E}\left[(\boldsymbol{\theta}^{mle})^3\right]$; these can be calculated using the skewness of the MLEs: $\mathrm{Skew}(\theta_i^{mle}) = 2/\sqrt{m}$, for $i = 1, 2$. The delta approximation of bias due to input modelling, b^{approx}, is a second-order approximation and therefore does not take the higher moments into account. However, in Table 4.4.3 it can be seen that as m increases $b^{approx} \to b$ since $2/\sqrt{m} \to 0$ as $m \to \infty$.

The first cubic function considered, Equation (4.4.5), was selected such that the quadratic approximation is reasonable over the space covered by the CCD design. In Table 4.4.3 we see that, when the smallest values of m and r were used, the lack-of-fit test is passed approximately the same proportion of times as the quadratic function, and we see similar results to the quadratic experiment. As m and r increase we see the power get increasingly close to 0.8 and the delta approximation, b^{approx}, converges to b. Overall our method works well for this example.

The second cubic function, Equation (4.4.6), was chosen so the quadratic assumption was a poor approximation over the CCD space for the smallest values of m and r considered. When $m = 10$ and $r = 50$ the lack-of-fit test rejected the the quadratic model in approximately 60% of the $G = 1000$ macro replications; this was the best case, but overall this test was not very sensitive to the lack of fit. In Table 4.4.3 we see that the power of the bias test is often higher than our nominal value of 0.8 for small values of m and r even when the average estimated bias, $\bar{\widehat{b}}$, differs substantially from b and b^{approx}; this is good, but we should not expect it to be a general phenomenon. Increasing m or r has the effect of shrinking the width of the CCD making the quadratic assumption over our CCD space a better approximation.

This experiment shows the importance of the locally quadratic assumption over the CCD space. When the quadratic assumption does not hold our estimate of bias, $\widehat{b}$, can be quite different from b when m is small. Using the lack-of-fit test to validate the quadratic assumption is therefore advised, but is not a panacea; recall this requires no additional simulation effort. Another problem is that, for small m, the distance between $\boldsymbol{\theta}^{mle}$ and $\boldsymbol{\theta}^{c}$ may be quite large, implying that we estimate the Hessian of the response surface at the wrong point which could impact both the estimate of bias and the power of the test.

4.4.4 A realistic example - NHS 111 healthcare call centre

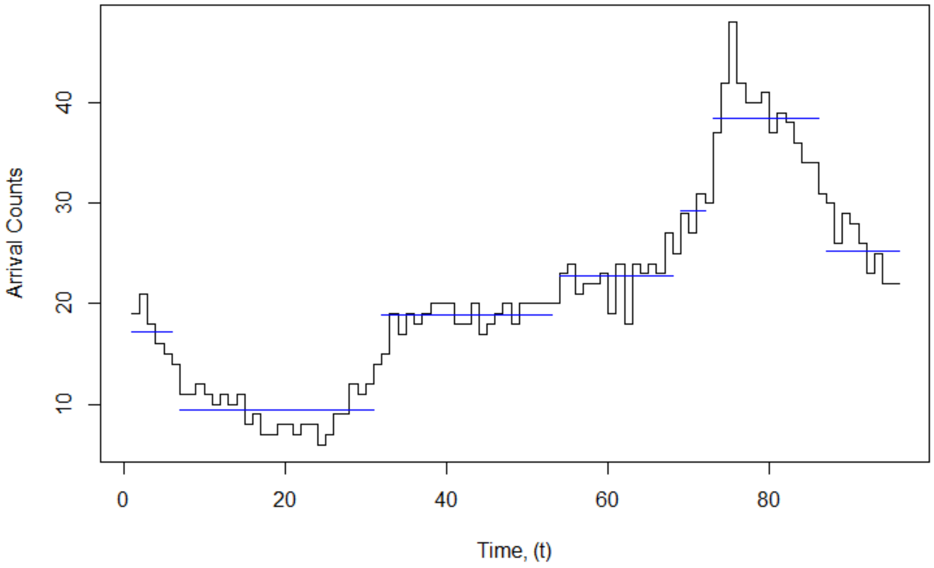
We now illustrate our bias detection diagnostic on the simulation of a real-world system with a non-stationary input process. The nominal experiment is based on observations of arrival counts over 96, 15-minute intervals, from an NHS 111 healthcare call centre in the UK. This system was introduced in Chapter 3 of this thesis. As previously described, the NHS 111 healthcare call centre system was designed to remove some of the strain from other healthcare services, for example emergency departments, by advising callers on which service they should access. Of the 6 months of data we had we decided to consider Wednesdays only as UK public holidays mid-week are rare and therefore we would expect no outliers in the arrival rates.

After checking the Poisson assumptions were satisfied by the arrival data, this system was simulated as an $M(t)/G/S(t)$ queueing model with a piecewise-constant Poisson arrival process. Based on data from the NHS 111 healthcare call centre system we conducted two experiments with different levels of input data. Let m_d denote the number of days the system was observed observed, and m denote the total number of arrivals over the m_d days. Figures 4.4.4a and 4.4.4b show the average rates over m_d =10 and m_d =26 days of arrival count data, respectively. In both scenarios change-point analysis for Poisson data, as discussed in Chen and Gupta (2011), was used to distinguish between intervals with significantly different arrival counts. This pre-processing technique was used because the IU in each small interval may be large, especially in intervals with low arrival rates where we would not expect to observe many arrivals. The change-point analysis reduced the arrival process to 7 and 8 intervals of varied length for the two scenarios; see the blue intervals in Figures 4.4.4a and 4.4.4b. Using the methods discussed in Chapter 3 we were then able to estimate the total IU, $\widehat{\kappa}$, of the expected waiting time of callers, $\mathrm{E}\left(\mathrm{WTime}\right)$, in both cases.

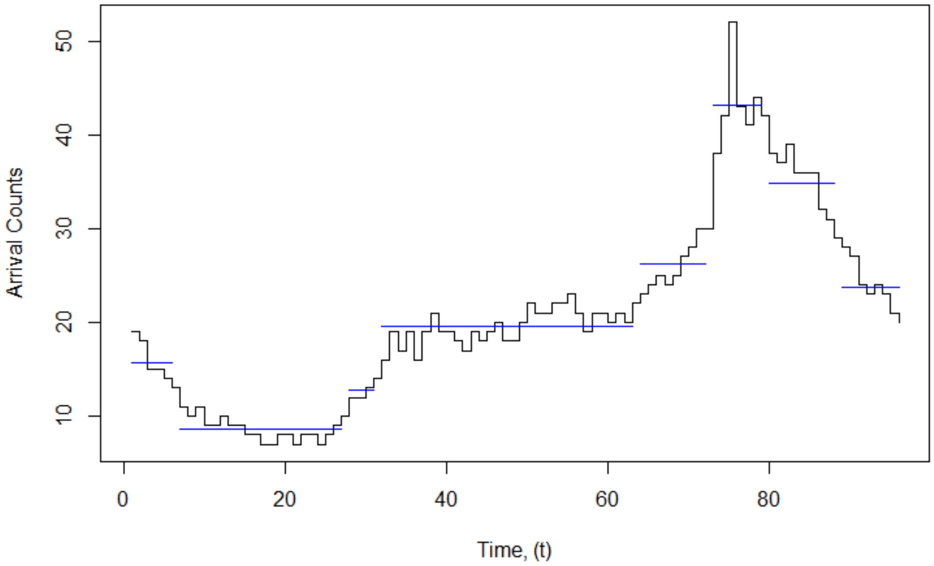
From two months of service-time data the mean service time was 8.00 minutes and the standard deviation was 4.33 minutes. A moment matching approach was used to fit a Gamma distribution with shape parameter $\phi_1 = 3.408$ and scale parameter $\phi_2 = 2.347$. Since we wanted to mimic having observed a service time for each arrival, we created a synthetic "observed" data set of service-time observations of size m corresponding to the expected number of arrivals in each scenario, and treated this as the real-world data.

To generate a realistic scenario we used approximately proportional staffing to meet the NHS target level of service, $\mathrm{P}(\mathrm{WTime} > 1 \text{ min}) < 0.05$. This corresponded to server utilisation of 62% in the model with 10 days of arrival data and 65% in the system with 26 days of arrival data. In the nominal experiment estimates of the expected waiting time of callers were found to be $\mathrm{E}(\mathrm{WTime}) = 0.0756$ minutes and $\mathrm{E}\left(\mathrm{WTime}\right) = 0.0674$ minutes respectively; this is our performance measure of interest.

In the experiments we carry out the bias diagnostic test, as described in §4.3, and within this we run the lack-of-fit diagnostic test to validate our quadratic approximation. An estimate $\widehat{\kappa}$ of IU variance is used to guide our choice of the relevant bias, γ. Note that, γ will therefore reduce with m, the number of arrival observations, because IU is also reduced. We want high power of rejecting the null if the true bias is larger than $\gamma = \sqrt{\upsilon \times \widehat{\kappa}}$ where $0 < \upsilon < 1$. This gives us a threshold of bias deemed to have an important effect on the



(a) The arrival count function given $m_d = 10$ days of observations.



(b) The arrival count function given $m_d = 26$ days of observations.

Figure 4.4.4: The average arrival counts over 96, 15 minute, intervals given m_d days of arrival data. Intervals post pre-processing of the data using change-point analysis are shown in blue.

MSE. Estimates of $\boldsymbol{\theta}^c$ and Ω were obtained from the input data, and σ^2 from the nominal experiment.

The desired power of the bias detection test was set equal to $1-\alpha_2 = 0.8$ and the size to $\alpha_1 = 0.05$; the size for the lack-of-fit test is also 0.05. For these experiments the relevant bias, γ, was set using $\upsilon = 0.3$, meaning we consider bias squared higher than 30% of the value of IU to be relevant.

For these two scenarios the number of input parameters driving the simulations are $k = 9$ and $k = 10$, respectively. This comes from the piecewise-constant arrival process having 7 or 8 distinct intervals, which are treated as independent input distributions; the final two parameters describe the service-time distribution. We conducted experiments employing both the full-factorial CCD and the reduced fraction CCD design proposed by Sanchez and Sanchez (2005). The latter design reduced the number of factorial points in both experiments to $n_F = 128$ from $n_F = 512$ and $n_F = 1024$ respectively. Note that in all experiments we repeat the centre point $n_C = 20$ times. The results of the bias detection test are displayed in Table 4.4.4.

Table 4.4.4: The bias detection test in a NHS 111 healthcare call centre scenario considering the expected waiting time of callers, $\mathrm{E}\left(\mathrm{WTime}\right)$, with $m_d = 10$ and $m_d = 26$ days of arrival data. Results for both the bias and the lack-of-fit tests are presented.

Design	Exp	m_d	m	n	r	γ	a	$\widehat{b}$	Bias	LOF
Full	1	10	20068	550	500	0.0035	0.577	0.0014	Accept	Reject
				550	1000	0.0035	0.485	0.0019	Accept	Accept
Frac	2	10	20068	166	500	0.0035	0.603	0.0013	Accept	Reject
				166	1000	0.0035	0.507	0.0005	Accept	Accept
Full	3	26	52711	1064	500	0.0024	0.699	0.015	Reject	Reject
				1064	1000	0.0024	0.588	0.011	Reject	Reject
Frac	4	26	52711	168	500	0.0024	0.737	0.005	Reject	Accept

Before we analyse the results of our bias detection test note that in Table 4.4.4 for experiments 1, 2 and 3 the result of the lack-of-fit test in the initial experiment with $r = 500$ replications at each design point was to reject the quadratic model. For this reason we repeated these experiments, increasing the number of replications at each design point from $r = 500$ to $r = 1000$. This did not change the conclusion of the bias detection test, but did result in experiments 1 and 2 passing the lack-of-fit test. Thus, in these two experiments with $r = 1000$ we have no strong evidence that our quadratic approximation is inadequate. In experiment 3, even with $r = 1000$, the lack-of-fit test rejects the null, suggesting a more complicated model is required to describe the response surface. Note that, although we doubled the number of replications at each design point the scaling factor of the design, a, only decreased by a small amount. Acquiring a scaling factor small enough for the quadratic approximation to hold may take a much larger number of replications; recall that $r = O(1/a^4)$.

In experiments 1 and 3 we use the full-factorial CCD and in experiments 2 and 4 we use the reduced fractional CCD by Sanchez and Sanchez (2005). In Table 4.4.4 we see that the conclusion of the bias detection test given the full CCD agrees with the conclusion when the reduced fractional design is used for both levels of arrival data. The scalability of our method was an issue of concern to us. Here we see a great reduction in the number of design points, n, and thus computational effort, required to estimate the bias due to input modelling when using the reduced experimental design, yet we are still able to gain an estimate $\widehat{b}$ close to the estimate from the full CCD and make the same conclusion.

In Table 4.4.4 we also see that, given a larger number of days of observations of the NHS 111 system γ, our relevant value of bias, decreases from $\gamma = 0.0034$ to $\gamma = 0.0024$. This is because we used IU to guide our value of γ and the estimate of IU, $\widehat{\kappa}$, is smaller in the system with more days of input data. Our bias detection test is set up so that when $|b| \geq \gamma$ we have high power of detecting bias. Since γ is higher in experiments 1 and 2 with $m_d = 10$ days of observations we require a larger departure from H_0 than we do in the experiments where $m_d = 26$ to have a high probability of rejecting the null. Further, given a large amount of input data the variability of the MLE's will be small. With our method this causes a smaller variance about the bias due to input modelling, $\text{Var}(\widehat{b})$, which in turn increases the power of our bias detection test.

Turning our attention to the conclusions of the bias detection tests in Table 4.4.4, we see that in experiments 1 and 2, with $m_d = 10$ days of arrival data, we accept the null hypothesis, so there is insufficient evidence to suggest $b \neq 0$ in these experiments. Since we set our threshold for relevant b^2 to 30% of the input uncertainty variance, and controlled the power to detect bias larger than this size, our conclusion is more practically stated as that bias is making a small contribution to overall MSE due to input modeling.

In experiments 3 and 4, with $m_d = 26$ days of observations, we reject the null hypothesis; that is, we have sufficient evidence to suggest that $b \neq 0$. At this point we may wish to spend additional computational effort on estimating $\widehat{b}$, to get a more accurate estimate of the bias due to input modelling about our performance measure estimate. Alternatively, at this point the practitioner may wish to reduce bias to a level that does not concern them by collecting more input data and repeating the bias detection test.

We have now illustrated our bias detection test on a realistic example. This example had a non-stationary piecewise-constant Poisson arrival process that we pre-processed using change-point analysis. Note that the location of the change-points will have had an effect on the bias due to input modelling. Change-point analysis aids the choice of arrival intervals but does not guarantee an arrival function that represents the true arrival process perfectly or that it propagates minimal error due to input modelling to our simulation output.

4.5 Conclusion

This chapter presents a test with controlled power for detecting bias due to input modelling of a relevant size in simulation models. Previously this form of error has been ignored. The test is built on the assumption that close to $\boldsymbol{\theta}^c$ the true response can be approximated by a quadratic. We fit the quadratic response surface using a CCD experimental design, which is constructed in a novel way allowing the practitioner to control the power of the bias detection test through the scaling of CCD width or the number of replications at each design point.

We explored and evaluated the bias detection test using a controlled experiment investigating the functional form of the response surface, the amount of input data and the number of replications completed at each design point. This experiment highlighted the importance of the validity of our quadratic assumption over the CCD space for our power to hold and we were able to show that by increasing the number of replications of the experiment at each design point or the number of observations used to estimate our input models we achieved our target power. Also of influence was the distance between the estimated input model parameters, $\boldsymbol{\theta}^{mle}$, and the true input model parameters, $\boldsymbol{\theta}^{c}$, which was seen to affect both the estimate of power and average bias estimate. We also demonstrated the bias detection test in a NHS 111 healthcare call centre example. This included the use of IU to guide our choice of the relevant value of bias.

From our exploration of quantifying and detecting bias due to input modelling there still remain open questions that may be of interest. One of these is how we might optimally set n_C the number of centre points in our model? Currently n_C is set in an ad hoc manner dependent on the number of factorial and axial points in the CCD. Also of interest is how we might optimally set r, the number of replications of the simulation at each design point. We need r large enough to ensure our quadratic assumption holds sufficiently closely but do not wish to waste unnecessary simulation budget. In the experiments in this chapter we chose r to be suitably large to satisfy our quadratic assumption. But another possibility could be to use the estimate of simulation noise, σ^2, from the nominal experiment to guide this choice.

In the NHS 111 example we used change-point analysis to form the arrival-process input model, which introduces its own error, but more generally input model misspecification is a source of model risk not captured here (e.g., if the arrival process is not actually Poisson). Similarly, we found that the lack-of-quadratic-fit test was not as strong an indicator as one might like of approximation error; this is an important problem for future study.

Note that our method can be used alongside current IU quantification techniques, allowing us to express the total error due to input modelling of our performance measures of interest. Current techniques allow IU quantification for simulation models with time-homogeneous arrival distributions and piecewise-constant non-stationary Poisson processes. Estimation and detection of error due to input modelling in simulation models with more complex arrival processes is something we leave for future work.

In conclusion, this chapter offers the first method for estimation and detection of bias due to input modelling. In doing so it allows a practitioner to consider the total error due to input modelling that may impact their performance measures of interest.

A Spline Function Method for Modelling and Generating a Nonhomogeneous Poisson Process

5.1 Introduction

Simulation models aim to mimic real-world systems and should therefore be driven by input models that represent well the behaviour of the system of interest. In this chapter we present a spline-based input modelling method with the aim of recovering the arrival rate of a nonhomogeneous Poisson process (NHPP) better than existing techniques in terms of both bias and variability; in doing so, we also reduce the input modelling error passed to the simulation output. Quantifying the error propagated to the simulation output caused by input modelling has been an active area of research in recent years, see Morgan et al. (2016), Morgan et al. (2017) and references therein.

In reality, many systems exhibit non-stationary behaviour, an example being arrivals to emergency departments which are known to be affected by the time of day and day of the week; see §5.5.2. A natural way to represent such behaviour is to use a NHPP. For a NHPP the rate, or intensity, $\lambda(t)$, is non-negative, $\lambda(t) \geq 0$, for all t, and is allowed to change through time. Given two points a and b, with $a \leq b$, let $N(a,b)$ denote the number of arrivals on the interval $(a,b]$. Note that, for a Poisson process the number of arrivals that occur in disjoint intervals are independent of one another. By the definition of a Poisson process $N(a,b)$ follows a Poisson distribution, $N(a,b) \sim \mathrm{Pois}(\Lambda(a,b))$, where $\Lambda(a,b)$ is known as the integrated rate, or cumulative intensity, function defined by $\Lambda(a,b) = \int_a^b \lambda(t)dt$. The probability of s arrivals occurring on interval $(a,b]$ from a NHPP with arrival rate function $\lambda(t)$ is $\mathrm{P}(N(a,b) = s) = \exp\{-\Lambda(a,b)\}\ \Lambda(a,b)^s/s!$. Since the probabilistic behaviour of a NHPP can be completely characterised by its rate function, $\lambda(t)$, or integrated rate function, $\Lambda(t)$, any input modelling approach for a NHPP therefore aims to estimate one of these functions. In this contribution we focus on estimation of the rate function $\lambda(t)$. For methods to estimate the integrated rate function see Leemis (1991) and Arkin and Leemis (2000) and references therein.

There are existing approaches for estimating the intensity function of NHPPs. Some of these make assumptions about the structure of the underlying rate function which limits their usefulness for modelling general processes. As an alternative, we propose a spline function arrival rate model. Spline functions are piecewise polynomials that are, by design, smooth and satisfy continuity constraints at the knots joining their pieces. In addition spline functions are flexible, becoming more so as the number of knots is increased. In this chapter we propose using a large number of knots allowing the resulting model to be very flexible.

The flexibility of the spline function enables a reduction in the bias between the input model and the true rate function, but flexibility can also lead to overfitting of the observed data. To control overfitting, and thus reduce the variability of the representation, we work with the penalised log-likelihood, adding a penalty parameter to the NHPP log-likelihood. For a fixed penalty value, we maximise the penalised log-likelihood of the NHPP using a trust region optimisation approach. Our method then selects the combination of spline coefficients and penalty by minimising a modified AIC score, known as the regularised information criterion (RIC), which accounts for the penalty in our penalised log-likelihood, see (Dixon and Ward, 2018) and references therein. The combination with the lowest RIC score is chosen.

Using the definition of a spline function as a linear combination of n B-spline basis functions we present a simple method for the simulation of arrivals from the NHPP via thinning. Using the decomposition property of a NHPP, the arrivals from the NHPP with arrival rate function represented by the spline-based model are the superposition of the arrivals simulated from the n spline components.

The chapter is organised as follows. In §5.2 we discuss the current literature for modelling the rate function of a NHPP. In §5.3 the spline-based input model is presented, and in §5.4 we introduce a thinning-based method for simulating arrivals from it. In §5.5 we evaluate our method in comparison to relevant competitors, present a realistic example of fitting a NHPP arrival rate function to arrivals from a real-world emergency department and consider how robust the spline-based method is to departures from Poisson data in terms over under- and overdispersion. In §5.6 we conclude.

5.2 Background

There are a number of NHPP input modelling techniques that utlise observed arrival times. The alternative for input modelling is to work with the counts of arrivals over intervals, see Nicol and Leemis (2014) and references therein. Note that, arrival-time observations can easily be transformed to arrival counts but counts cannot be transformed into arrival times. In this chapter we focus on arrival-time observations. A common approach to modelling arrival-time observations is to use an exponential form, $\lambda(t) = \exp\{g(t)\}$, where $g(t)$ is composed of additional polynomial or trigonometric components, as the exponential form ensures the rate function is always non-negative. This idea was adopted by Lewis and Shedler (1976), Lewis (1971), Kuhl et al. (1995), Kuhl et al. (1997), Lee et al. (1991) and others. Note that numerically optimising the parameters in these methods is computationally expensive and often requires a good starting point.

Other approaches assume the rate function is a piecewise polynomial of some degree. For example, Chen and Schmeiser (2013) present the iterative mean-constrained algorithm I-SMOOTH that returns a smoother piecewise-constant estimator of the arrival rate function given an initial piecewise-constant representation. Henderson (2003) shows that, when the intervals are of equal length, piecewise-constant estimators of the rate function are consistent as the number of arriavls increases, provided the length of the intervals shrink at an appropriate rate. Zheng and Glynn (2017) assume that the true intensity is piecewise-linear over known intervals and develop a convex programming formulation to estimate the intensity at the interval boundaries given arrival times or counts. Alternatively, Chen and Schmeiser (2017) present the Max Nonnegativity Ordering–Piecewise-Quadratic Rate Smoothing (MNO-PQRS) algorithm that produces a piecewise-quadratic representation of general input processes, not restricted to Poisson, over known intervals. Like I-SMOOTH, the MNO-PQRS algorithm is initialised with a piecewise-constant rate function. Kao and Chang (1988) present a piecewise polynomial representation given either arrival times or counts, where the breakpoints and polynomial degree in each interval are selected subjectively.

We now present a spline-based input modelling method for estimating the arrival rate function of a NHPP given arrival-time observations. Known uses for spline functions include: interpolation of data, approximate solutions of differential equations, curve approximation and image processing. Channouf (2008) uses a spline function to represent the rate function of both NHPPs and doubly stochastic Poisson processes. Unlike our approach, they do not make use of the B-spline composition of a spline function.

5.3 Fitting a spline function via penalised log-likelihood

Suppose we observe a NHPP with true rate function $\lambda^c(t)$, on the interval $[0,T]$, m_d times. In this chapter we let m_d be a number of days, but note in practice m_d could also represent other units such as minutes or hours or months. For flexibility we represent the rate function using a cubic, degree $e = 3$, spline function. A cubic spline function is a linear combination of n cubic basis functions, otherwise known as cubic B-splines. B-splines are locally defined functions. Let $B_{k,\boldsymbol{s}_k}(t)$ denote the k^{th} cubic B-spline at time t defined over the ordered knot sequence $\boldsymbol{s}_k = \{s_{k-(e+1)}, s_{k-e}, \ldots, s_k\}$. For $t \in \{s_{k-(e+1)}, s_k\}$, a cubic B-spline is nonnegative and twice continuously differentiable; otherwise it is equal to 0. For $e > 1$, B-splines are composed recursively from lower degree B-splines using the following recurrence relation

$$B_{k,e,\boldsymbol{s}_k}(t) = \frac{t - s_{k-(e+1)}}{s_{k-1} - s_{k-(e+1)}} B_{k,e-1,\boldsymbol{s}_k}(t) + \frac{s_k - t}{s_k - s_{k-e}} B_{k+1,e-1,\boldsymbol{s}_{k+1}}(t), \tag{5.3.1}$$

for $t \in [s_{k-(e+1)}, s_k)$, where e denotes the degree of the B-spline. At the lowest level, $e = 0$, this is

$$B_{k,0,\boldsymbol{s}}(x) = \begin{cases} 1 & \text{if } s_{k-1} \leq x < s_k \\ 0 & \text{otherwise.} \end{cases}$$

Given the definition of a B-spline, the spline rate function is defined by

$$\lambda(t;\boldsymbol{c}) = \sum_{k=1}^{n} c_k B_{k,\boldsymbol{s}_k}(t), \tag{5.3.2}$$

where $c_k \in \mathbb{R}$ is the spline coefficient of the k^{th} B-spline and $\boldsymbol{c} = \{c_1, c_2, \ldots, c_n\}$. Note that, as n gets larger there are more B-splines, and thus more knots, resulting in increased flexibility of the shape of the spline function. Spline function (5.3.2) combines the n local knot vectors of its component B-splines. Let the knot sequence of the spline function be denoted $\boldsymbol{s}$, where $\boldsymbol{s} = \{s_{-e}, s_{-e+1}, \ldots, s_0, s_1, \ldots, \ldots, s_{n+1}\}$. It may seem unconventional to start knot sequence $\boldsymbol{s}$ with knot s_{-e} but, if we are interested in estimating an arrival rate function on the interval $[0, T]$, by setting $s_0 = 0$ and $s_{n-e} = T$ we ensure that for all $t \in [0, T]$, $e + 1$ B-splines are non-zero. In general, the knots of a spline function need not be uniformly spaced but we will focus on uniformly spaced knot vectors, also known as cardinal B-splines. Cardinal B-splines are horizontal translates of each other; in §5.4 we will discuss how this can be advantageous for arrival generation. From herein, we drop the knot sequence subscript on the B-spline and let $B_k(t)$ denote the k^{th} B-spline unless necessary.

Note that once the knots of the spline function have been placed the value of each B-spline is fixed for all t. The resulting spline rate function is completely determined by the spline coefficients, $\boldsymbol{c} = \{c_1, c_2, \dots, c_n\}$. In fitting the spline function it is therefore the spline coefficients we wish to optimise.

5.3.1 The penalised log-likelihood

We chose to fit the spline function given a large number of knots. This allows flexibility of the resulting spline function, but may lead to a representation, $\lambda(t;\boldsymbol{c})$, that over fits the observed data. To control this, when fitting $\lambda(t;\boldsymbol{c})$ we use a penalised log-likelihood. The likelihood of a NHPP conditional on m_d days of observations over the interval $[0,T]$ is

$$L(\lambda(t;\boldsymbol{c})) \propto \prod_{i=1}^{m_d} \prod_{j=1}^{a_i} (\lambda(t_{ij};\boldsymbol{c})) \exp\left\{ -\int_0^T \lambda(y;\boldsymbol{c}) dy \right\}_d^m$$

where a_i denotes the number of arrivals observed on the i^{th} day, and $0 \leq t_{i1} < t_{i2} < \cdots < t_{ia_i} \leq T$, $i = 1, 2, \dots, m_d$ denote the observed arrival times. This gives us the log-likelihood

$$l(\lambda(t;\boldsymbol{c})) \propto \sum_{i=1}^{m_d} \sum_{j=1}^{a_i} \log(\lambda(t_{ij};\boldsymbol{c})) - m_d \int_0^T \lambda(y;\boldsymbol{c}) dy.$$

We chose to penalise the log-likelihood using a measure of the curvature of the fitted rate function: the integrated second derivative of the spline function

$$\frac{1}{2} \int_0^T \{\lambda''(u;\boldsymbol{c})\}^2 du. \tag{5.3.3}$$

This is a standard penalty for cubic splines within the smoothing spline literature (de Boor, 1978). The penalised log-likelihood is thus

$$\begin{aligned} l_p(\lambda(t;\boldsymbol{c})) &\propto \sum_{i=1}^{m_d} \sum_{j=1}^{a_i} \log\left(\lambda(t_{ij};\boldsymbol{c})\right) - m_d \int_0^T \lambda(y;\boldsymbol{c}) dy - \frac{1}{2}\theta \int_0^T \{\lambda''(u;\boldsymbol{c})\}^2 du \\ &\propto \sum_{i=1}^{m_d} \sum_{j=1}^{a_i} \log\left(\sum_{k=1}^{n} c_k B_k(t_{ij}) \right) - m_d \sum_{k=1}^{n} c_k \int_0^T B_k(y) dy - \frac{1}{2}\theta \sum_{k=1}^{n} \sum_{h=1}^{n} c_k c_h \int_0^T B_k''(u) B_h''(u) du. \end{aligned} \tag{5.3.4}$$

where $\theta \in [0, \infty)$ is a penalty parameter. When $\theta = 0$ we return to the un-penalised log-likelihood of the NHPP. When θ is large it drives the penalty (5.3.3) down forcing the spline function $\lambda(t;\boldsymbol{c})$ to be smoother; in the limit as $\theta \to \infty$ the rate function becomes linear. Recall that, when the knots have been placed the value of $B_k(u)$ is fixed for all u, thus $\int_0^T \{B_k''(u)\}^2 du$ is fixed for $k = 1, 2, \ldots, n$.

For a fixed penalty θ, we optimise $\boldsymbol{c}$, by maximising the penalised log-likelihood using a trust region approach. We denote the optimised spline coefficients for a given penalty, θ, by $\widehat{\boldsymbol{c}}_\theta$. Later we will use an information criterion to select θ.

5.3.2 Trust region optimisation

Trust region optimisation, see Conn et al. (2000), makes use of a local model, $\tau(\cdot)$, of the function to be optimised, here the penalised log-likelihood, and iteratively steps closer to the optimal solution by taking steps within a region where $\tau(\cdot)$ is trusted. By convention the trust region approach is a minimisation algorithm, we therefore minimise the negative penalised log-likelihood as an equivalent to our problem. This requires a local approximation of the negative penalised log-likelihood, $-l_p(\cdot)$. As is typical in the trust region approach, we use a second-order Taylor series to describe the local model. In each iteration of the algorithm we centre the model on the current, fixed, estimate of our spline coefficients, $\boldsymbol{c}_h$

$$\tau(\boldsymbol{p}_h) = -(l_p(\boldsymbol{c}_h) + g_p(\boldsymbol{c}_h)^T \boldsymbol{p}_h + \frac{1}{2}\boldsymbol{p}_h^T H_p(\boldsymbol{c}_h)\boldsymbol{p}_h)$$

where $\boldsymbol{p}_h$ is the proposed vector of parameter values, g_p is the $(1 \times n)$ gradient vector and H_p is the $(n \times n)$ Hessian matrix of $l_p(\cdot)$, which can be calculated exactly.

Within the trust region algorithm we iteratively step towards the optimum within a region in which we believe model $\tau(\cdot)$ to be a good approximation of $-l_p(\cdot)$. For this reason the trust region approach imposes an upper limit on how large of a step we can take in each iteration. The area in which the algorithm is allowed to move is call the ‘trust region’. Let us denote the radius of this ‘trust region’, centred at $\boldsymbol{c}_h$, as Δ.

At each iteration, h, of the trust region algorithm we solve the trust region sub-problem:

$$\min\ \tau(\boldsymbol{p}_h) \qquad \text{subject to: } ||\boldsymbol{p}_h|| \leq \Delta \quad \text{and} \quad \boldsymbol{c} \geq \boldsymbol{0}, \tag{5.3.5}$$

where $||\cdot||$ denotes Euclidean distance. The constraint $\boldsymbol{c} \geq \boldsymbol{0}$ is used to ensure nonnegativity of the resulting rate function. Note that this constraint is stronger than necessary since negative spline coefficients are possible whilst still maintaining a positive rate function. But the constraint leads to a simple way to force the rate function, $\lambda(t;\boldsymbol{c})$, to stay nonnegative. The trust region subproblem (5.3.5) is a convex, quadratic program and thus has a unique solution. To solve it in practice we used the Gurobi Optimization (2018) quadratic solver.

Note that if the true rate function, $\lambda^c(t)$, is known to have a cyclic structure, we can impose this structure upon our spline function by adding constraints of the form

$$\lambda(0;\boldsymbol{c}) = \lambda(T;\boldsymbol{c}), \quad \lambda'(0;\boldsymbol{c}) = \lambda'(T;\boldsymbol{c}), \quad \lambda''(0;\boldsymbol{c}) = \lambda''(T;\boldsymbol{c}),$$

to the trust region subproblem. Such constraints can easily be incorporated.

Given a proposed step $\boldsymbol{p}_h$ from the trust region subproblem, we decide whether to accept or reject the step according to the ratio

$$\rho_h = \frac{l_p(\lambda(t;\boldsymbol{c}_h)) - l_p(\lambda(t;\boldsymbol{c}_h + \boldsymbol{p}_h))}{\tau(\boldsymbol{0}) - \tau(\boldsymbol{p}_h)}. \tag{5.3.6}$$

This ratio compares the actual reduction in the penalised log-likelihood to the predicted reduction from the model. A value of ρ_h close to 1 says that there is good agreement between model $\tau(\cdot)$ and $-l_p(\cdot)$. We accept $\boldsymbol{p}_h$ if $\rho_h > \alpha$, where α is set by the practitioner. If $\boldsymbol{p}_h$ is accepted our new position is $\boldsymbol{c}_{h+1} = \boldsymbol{c}_h + \boldsymbol{p}_h$.

Note that the radius of the trust region Δ is adaptive throughout the algorithm. If ρ_h is close to 1 and $||\boldsymbol{p}_h|| = \Delta$, then Δ is restricting our step, and we would increase the radius of the trust region. Alternatively, if there is not a good agreement between $\tau(\cdot)$ and $-l_p(\cdot)$, we restrict the model to the region where the Taylor series approximation is better. Rules on when to change the trust region radius are set using thresholds. For example, we might change Δ when $\rho_h < \gamma = 0.25$ or $\rho_h > \beta = 0.75$. As we get closer to the optimum, $\widehat{\boldsymbol{c}}_\theta$, the size of the trust region shrinks. We stop the algorithm when $||\boldsymbol{p}_h|| < \varepsilon$ where ε is a stopping value set by the user. Within the trust region approach α, β, γ and ε are decided by the practitioner; sensible values are suggested by Wright and Nocedal (1999).

The trust region algorithm is used within the spline-based method to minimise the negative penalised log-likelihood, and thus optimise the spline-coefficients for a fixed penalty, θ. One drawback of the trust region algorithm is that it may struggle to converge to the true optimal spline coefficient values or even stall when the number of knots grows too large. The dimension of the optimisation problem increases with the number of knots used to build the spline function, and thus finding the optimum is harder as the number of knots grows. We must also take into account that model $\tau(\cdot)$ is a second-order approximation, whereas the spline function is a cubic polynomial. As the number of knots increases the spline function becomes more and more flexible on smaller and smaller intervals. This means that to ensure model $\tau(\cdot)$ is a valid approximation at the point $\boldsymbol{c}_h$ we must take smaller steps, $\boldsymbol{p}_h$. The smaller the step we take in each iteration the slower the convergence and in some cases the algorithm may even stop before the optimum has been reached. Note that this is a problem in the trust region algorithm; if the spline coefficients do not converge to their optimal value for a chosen number of knots this does not mean that a spline function cannot be fit with that number of knots. It may be possible to adaptively change the parameters of the trust region algorithm, in the same way that the size of the trust region radius, Δ, changes, to ensure convergence occurs. Another possibility is to use a different optimisation approach to find the optimal spline coefficients, we leave this as suggested future work.

5.3.3 Selecting $\{\theta, \widehat{\boldsymbol{c}}_\theta\}$

To choose the combination of penalty parameter and spline coefficients, $\{\theta, \widehat{\boldsymbol{c}}_\theta\}$, we use a modification of the AIC score of Cavanaugh and Neath (2011), known as the regularisation information criterion (RIC); see Dixon and Ward (2018) and Shibata (1989). As with most information criteria, this score is based on Kullback-Leibler (KL) information, a measure of the distance between two distributions (Kullback, 1997). Both the AIC and RIC trade off the goodness-of-fit of a proposed model, in this case a spline function, and its complexity. If we selected the combination $\{\theta, \widehat{\boldsymbol{c}}_\theta\}$ by maximising the penalised log-likelihood alone we would always choose the unpenalised spline function, where $\theta = 0$, as it is more able to adapt the characteristics in the observed data. A penalty is therefore added to the penalised log-likelihood to control overfitting. Where degrees of freedom is used in traditional AIC, RIC uses the effective degrees of freedom, e, defined as follows

$$\begin{aligned}\text{RIC} &= -2\, l(\lambda(\boldsymbol{t};\widehat{\boldsymbol{c}}_\theta)) + 2\, e,\\ &= -2\, l(\lambda(\boldsymbol{t};\widehat{\boldsymbol{c}}_\theta)) + 2\, \text{tr}(I_p(\widehat{\boldsymbol{c}}_\theta) J_p(\widehat{\boldsymbol{c}}_\theta)^{-1}).\end{aligned}$$

Within the effective degrees of freedom, $I_p(\widehat{\boldsymbol{c}}_\theta)$ is the observed Fisher information and $J_p(\widehat{\boldsymbol{c}}_\theta)$ is the negative Hessian matrix of the penalised log-likelihood,

$$\begin{aligned}I_p(\widehat{\boldsymbol{c}}_\theta) &= \frac{1}{m_d}\sum_{i=1}^{m_d} \frac{\partial}{\partial \boldsymbol{c}}\left[l(\lambda(\boldsymbol{t}_i;\widehat{\boldsymbol{c}}_\theta)) - \theta\frac{\Omega}{2m_d}\right]\frac{\partial}{\partial \boldsymbol{c}'}\left[l(\lambda(\boldsymbol{t}_i;\widehat{\boldsymbol{c}}_\theta)) - \theta\frac{\Omega}{2m_d}\right]\\ J_p(\widehat{\boldsymbol{c}}_\theta) &= -\frac{1}{m_d}\sum_{i=1}^{m_d}\frac{\partial^2}{\partial \boldsymbol{c}\partial \boldsymbol{c}'}\left[l(\boldsymbol{t}_i;\widehat{\boldsymbol{c}}_\theta) - \theta\frac{\Omega}{2m_d}\right] = -H_p(\widehat{\boldsymbol{c}}_\theta),\end{aligned}$$

where $\boldsymbol{t}_i$ are the arrivals observed on the i^{th} day and Ω is the matrix of partial second derivatives of the penalty function, (5.3.3), $\Omega_{ij} = \int_0^T B_i''(u)B_j''(u)du.$ The chosen combination, $\{\theta,\widehat{\boldsymbol{c}}_\theta\}$, is composed of the values of the penalty parameter and spline coefficients that minimise the RIC. Given the penalty value θ, the optimal spline coefficients, $\widehat{\boldsymbol{c}}_\theta$, can be found using trust region optimisation as discussed in §5.3.2. This reduces the search for the combination $\{\theta,\widehat{\boldsymbol{c}}_\theta\}$ to finding the penalty value, $\theta \in [0,\infty)$ that minimises the RIC. This is a one-dimensional search: at each step we

1. Fix θ,
2. maximise the penalised log-likelihood to find $\widehat{\boldsymbol{c}}_\theta$,
3. then evaluate the RIC at $\{\theta, \widehat{\boldsymbol{c}}_\theta\}$.

For speed, we propose a simple search to narrow down the interval in which to select θ. We suggest starting with a high penalty value η and jumping backwards towards 0 by halving the penalty at each step, $\theta = \{\eta, \frac{1}{2}\eta, \frac{1}{4}\eta, \frac{1}{8}\eta, \dots\}$; this allows us to take larger steps initially. Note that, if in the first jump $\text{RIC}_\eta < \text{RIC}_{\frac{1}{2}\eta}$ then we would restart the algorithm from a higher starting point as we are moving in the wrong direction to find the minimum RIC. Let us say that in the q^{th} step we observe $\text{RIC}_{\frac{1}{2^q}\eta} < \text{RIC}_{\frac{1}{2^{q+1}},\eta}$ for the first time, then we know that the minimum must lie within the interval $O = \{\frac{1}{2^{q+1}}\eta, \frac{1}{2^q}\eta\}$. We have therefore narrowed the search for the penalty to a one-dimensional search within interval O. In practice we completed the one-dimensional search for θ in O using the R function optimise (R Core Team, 2018), which combines a golden section search and successive parabolic interpolation.

At this point we have provided a method to construct a spline-based input model, $\lambda(t;\widehat{\boldsymbol{c}}_\theta)$, for the arrival rate of a NHPP. Although not the topic of this chapter, it may also be of interest to consider the pointwise variability in the spline-based representation, $\lambda(t;\widehat{\boldsymbol{c}}_\theta)$. This could, for example, be used in the construction of pointwise confidence intervals. For some $t \in [0,T]$ the variance of the spline function $\lambda(t;\widehat{\boldsymbol{c}}_\theta)$, is

$$\text{Var}[\lambda(t;\widehat{\boldsymbol{c}}_\theta)] = \text{Var}[\widehat{\boldsymbol{c}}_\theta \boldsymbol{B}(t)] = \boldsymbol{B}(t)^T \text{Cov}[\widehat{\boldsymbol{c}}_\theta]\boldsymbol{B}(t) \tag{5.3.7}$$

where $\boldsymbol{B}(t) = \{B_1(t), B_2(t), \ldots, B_n(t)\}$ is the vector containing the value of each B-spline at time t. When the penalty function induces little bias on the estimates, Gray (1992) justifies estimating the variance-covariance matrix of the spline coefficients, $\widehat{\boldsymbol{c}}_\theta$, by

$$\text{Cov}[\widehat{\boldsymbol{c}}_\theta] = m_d H_p(\widehat{\boldsymbol{c}}_\theta, \theta^{(m)})^{-1} I(\widehat{\boldsymbol{c}}_\theta) H_p(\widehat{\boldsymbol{c}}_\theta, \theta^{(m)})^{-1},$$

as $m_d \to \infty$, where $I(\widehat{\boldsymbol{c}}_\theta)$ denotes the observed information matrix of the unpenalised log-likelihood and $\theta^{(m_d)}$ is a sequence of penalty values such that $\theta_j^{(m_d)}/m_d \to Q_j$, where $0 \leq Q_j < \infty$, as $m_d \to \infty$. This sequence is used to achieve the same degree of smoothing as m_d increases since the contribution of the log-likelihood to the total penalised log-likelihood increases with m_d. When the bias induced on the estimates of the spline coefficients by the penalty is large, formal inference about the error in $\widehat{\boldsymbol{c}}_\theta$ is not advised (Gray, 1992). In §5.5.2, under the additional approximation of normality, we use (5.3.7) to estimate a 95% pointwise confidence interval around the spline-based model.

5.4 Generating arrivals from the spline function

At this point we have presented a spline function method for fitting the rate function of a NHPP. In practice we wish to be able to generate arrivals from this function to drive our simulation models. Given we directly model the rate function, thinning is arguably the most appropriate method for arrival generation in this context. To generate arrivals using the thinning method, the maximum of the intensity function, or at least some majorising function, is required. For the spline function representation, Equation (5.3.2), the maximum is not straightforward to calculate, but we do know the maximum of each B-spline function.

The composition of the spline function representation is advantageous for arrival generation. By the superposition property of NHPPs (Kingman, 1992), it is known that the sum of n independent NHPPs is also a NHPP. When this is the case, the intensity function of the process is the sum of the intensity functions of its n components, $\lambda^c(t) = \lambda_1^c(t) + \lambda_2^c(t) + \cdots + \lambda_n^c(t)$. Given the form of (5.3.2), it would therefore be natural to treat each component of the spline function as the intensity of an individual NHPP, $\lambda_k(t) = c_k B_k(t)$, for $k = 1, 2, \ldots, n$. The key advantage being that the maximum of the B-spline basis functions are known.

Each cubic B-spline is built on a local knot sequence of 5 knots, for the k^{th} B-spline this is $\{s_{k-4}, s_{k-3}, s_{k-2}, s_{k-1}, s_k\}$. The maximum of each cubic B-spline function is known to fall at the centre of its local knot sequence, $B_k(s_{k-2})$ for the k^{th} B-spline. Since this maximum is known, we can also calculate the maximum of $\lambda_k(t) = c_k B_k(t)$, for $k = 1, 2, \ldots, n$. Using thinning we can generate arrivals from each component NHPP; the superposition of these arrival times are the arrivals from the NHPP with intensity $\lambda(t;\boldsymbol{c})$ as required. When cardinal B-splines are used as the basis for the spline function, arrival generation simplifies even further as all spline components, $\lambda_k(t;\boldsymbol{c})$, are simply a scaled translation of the first B-spline, $\lambda_k(t;\boldsymbol{c}) = c_k B_1(t)$, with maximum at knot s_{-1}.

When generating arrivals from each spline component, we also propose using the knowledge of the maxima of each component to create a tight, piecewise-linear majoris-

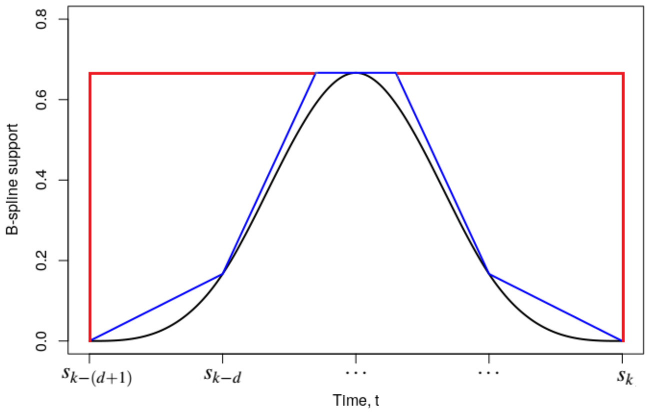


Figure 5.4.1: The k^{th} spine function component plus, a piecewise-constant majorising function (red) and an example of a piecewise-linear majorising function (blue) that could be used to generate arrivals via inversion using the method of Klein and Roberts (1984).

ing functions for use within the thinning algorithm. Klein and Roberts (1984) propose a simple method for generating arrivals from a piecewise-linear function. When the arrival rate is piecewise-linear, the integrated rate function is piecewise-quadratic, and Klein and Roberts (1984) provide a tractable form for the inverse of the integrated rate function and an algorithm for efficient arrival generation via inversion. Figure 5.4.1 illustrates the constant majorising function and an example piecewise-linear majorising function for the k^{th} spline component, $\lambda_k(t;\boldsymbol{c}) = c_k B_k(t)$, where $c_k = 1$. The tighter the fit of the majorising function the more efficient thinning will be. It is clear from Figure 5.4.1 that the piecewise-linear majorising function provides a tighter fit than the constant function. An indicator of how tight the fit of the majorising function is, is the ratio of the area under the B-spline to the area under the majorising function, where 1 is perfect agreement. For the constant majorising function this ratio is 2.67, and for the piecewise-linear majorising function it is 1.14. Of course we could reduce this ratio further by more careful selection of the piecewise-linear majorising function, but we leave this to further work. In terms of efficiency, the more arrivals that need generating within a simulation the more important it will be to reduce the gap between the spline component and its majorising function.

5.5 Evaluation

In this section we evaluate our spline function input modelling method by comparing it to two input modelling methods that have recently been presented in the literature. We also illustrate the use of fitting a spline function to observations of arrivals of a real-world accident and emergency (A&E) department and investigate the robustness of the spline-based method for fitting the arrival rate function of an input process when the observations are under- or overdispersed in comparison to a Poisson process.

5.5.1 Computational Comparison

We start the evaluation by comparing our spline-based method to two appropriate competitors in the existing literature, the piecewise-quadratic input model presented by Chen and Schmeiser (2017), known as MNO-PQRS, and the piecewise linear approach by Zheng and Glynn (2017). Here, by an "appropriate" method we mean methods that are able to take arrival-time observations from a NHPP and fit the rate function whilst making no prior assumptions about the trends of the underlying rate function.

In the following experiments the true rate function, $\lambda^c(t)$, on the interval $[0,24]$, is made up of two components: a sinusoidal function that affects the whole interval and a peak, constructed using the density function of a normal distribution, which only affects part of the interval. Constructing the peak using a normal density allows manipulation of the height and the length of effect of the peak on the arrival rate. In this way we are able to test how well each of the methods estimate both abrupt and slow changes to the rate function over time. Let κ denote the how many times higher the peak is at its mid-point than the underlying sinusoidal function at that point, and ξ denote the approximate duration of the peak. We consider values of $\kappa = 1,\ 3$ and 5 and $\xi = 1,\ 5$ and 10 and the mid-point of the peak is always at $t = 15$. Also of interest is the number of observations of the system. We will consider having $m_d = 15,\ 30$ and 100 sets of observations over the interval [0,24] from which to fit the arrival rate.

In each experiment $G = 500$ sets of m_d observations are simulated from each rate function given κ and ξ; this leads to $G = 500$ representations of the rate function for each method. To compare the methods on the interval $[0,T]$ we observe the integrated absolute difference, δ, and the maximum absolute difference, ζ,

$$\delta = \int_0^T |\widehat{\lambda}(q) - \lambda^c(q)|\, dq$$
$$\zeta = \max_{0 \leq q \leq T} |\widehat{\lambda}(q) - \lambda^c(q)|.$$

in each replication. These metrics are indicators of how well the estimated rate function recovers the truth, $\lambda^c(t)$. We record the average integrated absolute difference and average maximum absolute gap over the $G = 500$ replications denoted $\bar{\delta}$ and $\bar{\zeta}$ respectively. We also record the coefficient of variation of the integrated absolute difference, denoted ι, as an indicator of the dispersion of the fit of each method. This allows us to comment on the variability, or stability, of the methods.

Both competing methods are piecewise, and assume the number and position of the intervals are known. In this experiment the true functions are not piecewise, this information is therefore unknown; we choose to pre-process the simulated data in each of the $G = 500$ replications using the method presented by Chen and Schmeiser (2018). This method is data driven, using the mean integrated squared error (MISE) to choose an optimal number of equal-length intervals. The interval placement was therefore the same for the piecewise-linear and piecewise-quadratic methods in each replication. For the spline-based method, 50 equally spaced knots were used.

In total 9 arrival rate functions were considered for three levels of input data totalling 27 experiments for each modelling method. The methods are denoted “SPL”, “PQ” and “PL”, respectively.

In all 27 experiments the spline-based method out-performed the piecewise-quadratic and piecewise-linear approaches by attaining the lowest average integrated absolute difference and the lowest average maximum absolute difference. In all but a small number of experiments the spline-based method also attained the lowest coefficient of variation of the integrated absolute difference indicating a higher level of stability than the competing methods. Given the promising results from all 27 experiments we chose two extreme cases to present, one rate function with a short duration but a high peak and one with a long duration and low peak. In Table 5.5.1 we see that for both experiments the spline-based method has an average integrated absolute difference considerably lower than its competitors. It also has a lower dispersion index, ι, which indicates that the spline fit is more stable than the other methods and that the integrated absolute difference does not stray far from the average over the $G = 500$ replications; this indicates that the penalisation of the likelihood works as intended.

For both experiments presented in Table 5.5.1, we also plotted a single fit of the arrival rate function using the three input modelling methods; to be specific the chosen fitted arrival rate functions were from the replication where the spline function achieved its maximum maximum absolute gap, minimum maximum absolute gap, maximum absolute integrated difference and minimum absolute integrated difference. The arrival rate functions are presented in Figures 5.5.1 and 5.5.2. In plotting these figures we see how the spline function competitors perform when the spline function performs best and worst. In Figure 5.5.1 we see the fit of the three methods to the arrival rate function of an NHPP with an abrupt, $\xi = 1$, high magnitude, $\kappa = 5$, peak given $m_d = 15$ sets of observations. Due to the abrupt peak all three methods struggle to fit this function. It is clear that when the spline-based

Table 5.5.1: The average maximum absolute difference, $\bar{\zeta}$, the average integrated absolute difference, $\bar{\delta}$, and the coefficient of variation of the integrated absolute difference, ι, for the fit of two arrival rate functions given m_d, κ and ξ.

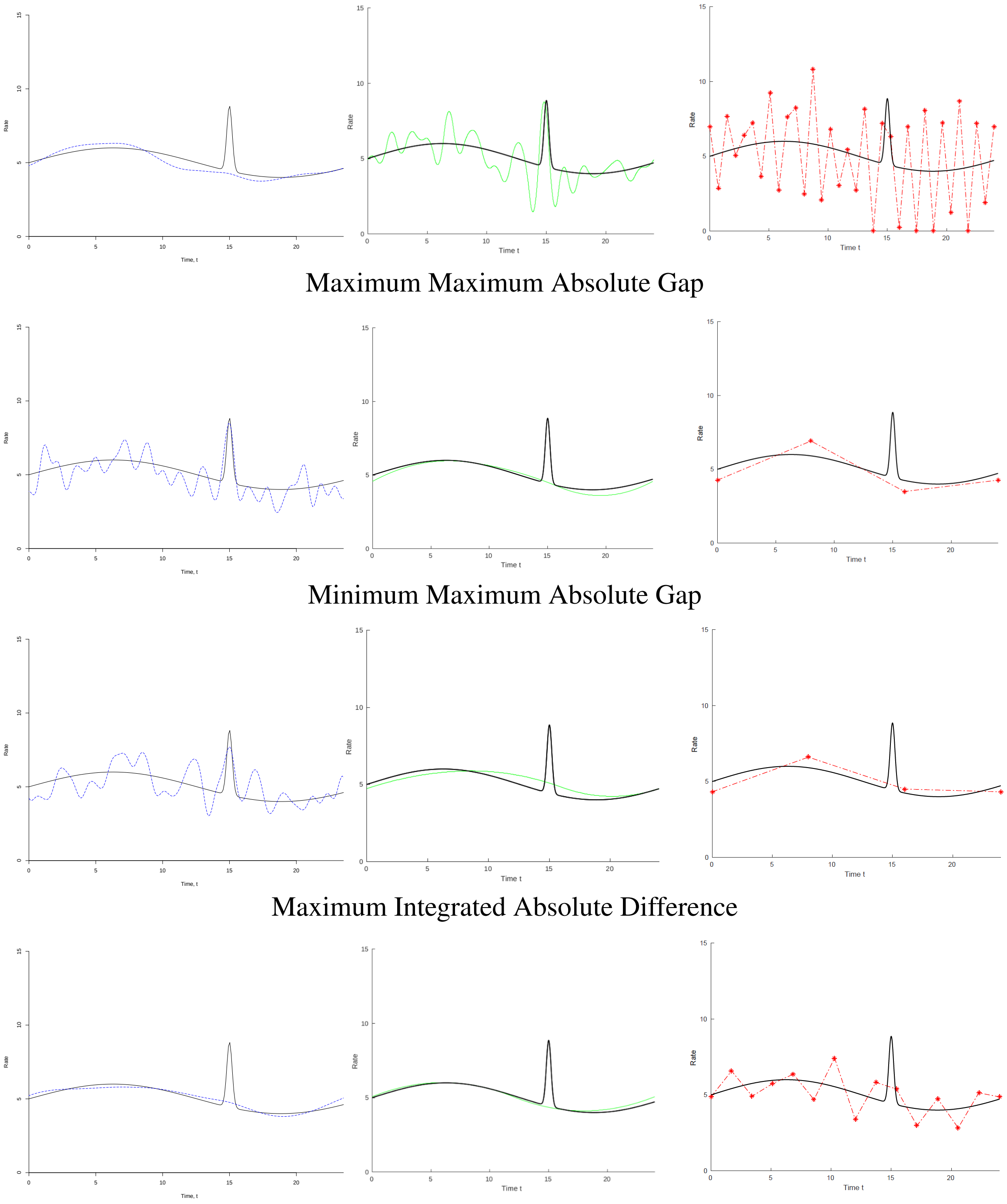
	$m_d = 15$, $\kappa = 5$, $\xi = 1$			$m_d = 100$, $\kappa = 1$, $\xi = 10$		
	$\bar{\zeta}$ (se)	$\bar{\delta}$ (se)	ι	$\bar{\zeta}$ (se)	$\bar{\delta}$ (se)	ι
SPL	3.78 (1.92×10^{-2})	7.71 (8.99×10^{-2})	0.26	0.30 (3.66×10^{-3})	2.54 (2.92×10^{-2})	0.26
PQ	4.15 (2.40×10^{-2})	11.30 (1.79×10^{-2})	0.35	1.74 (1.35×10^{-2})	3.74 (5.56×10^{-2})	0.31
PL	4.11 (3.99×10^{-2})	18.62 (3.65×10^{-1})	0.44	1.04 (1.80×10^{-2})	7.17 (1.24×10^{-1})	0.39

method estimates the peak well it becomes erratic elsewhere and when it fits the underlying sinusoid well it smooths over the peak completely. In this case the two metrics, maximum gap and integrated absolute difference, oppose each other; the spline-based method with the smallest maximum gap occurs when the spline estimates the peak well but the smallest integrated absolute difference occurs when the peak is ignored completely. When m_d is small there appears to be no pattern between the arrival rate fit by the spline function and the other two methods. In most cases, it appears that the piecewise-quadratic and piecewise-linear methods have been fit given a small number of intervals from the pre-processing algorithm. It is clear that the flexibility of both methods is greatly affected by the choice of the number of intervals. This indicates that prior knowledge of the number and placement of intervals is important to both methods. The spline function, on the other hand, was given 50 equally spaced knots in all replications.

In Figure 5.5.2 we see the fit of the three methods to the arrival rate function of an NHPP with a long, $\xi = 10$, small magnitude, $\kappa = 1$, peak given $m_d = 100$ sets of observations. In this example the behaviour of all three methods is similar; note, the number of pieces suggested by the preprocessing technique was higher. The fit of the rate function is good in all cases, but when the spline performs less well, for example in the case of the maximum maximum absolute gap and maximum integrated absolute difference, the PQ and PL methods also perform less well.

In addition to Figures 5.5.1 and 5.5.2, for the same two experiments we also plotted pairwise comparisons of the methods for both metrics over the full $G = 500$ fits, see Figures 5.5.3-5.5.6. These plots allow us to see any pairwise-correlation in performance between the methods over all replications. In Figures 5.5.3 and 5.5.4 we consider the arrival rate with an abrupt peak given $m_d = 15$ days of observations. For both metrics we can see that, in the majority of replications, the spline-based method is more stable and performs better than both of its competitors. This is more pronounced in comparison to the piecewise-linear approach. In comparing the piecewise-linear and piecewise-quadratic methods, the piecewise-quadratic method appears to perform better in terms of the integrated absolute difference in most cases; both methods perform similarly in terms of the maximum absolute gap. In Figures 5.5.5 and 5.5.6 we consider the arrival rate with a long duration, short peak given $m_d = 100$ days of observations. It is clear that the spline-based method performs best in the majority of experiments when considering the integrated absolute difference and in all experiments when considering the maximum absolute gap. Again the spline-based method is more stable than its competitors in terms of both metrics. In this experiment when comparing the piecewise-quadratic and piecewise-linear methods the piecewise-quadratic method appears to perform best in terms of the integrated absolute difference, and the opposite appears to be true for the maximum absolute gap. The conclusion from Figures 5.5.3-5.5.6 is that it does not appear that the methods perform their best or worst at the same time, and it is clear that the spline based method is out-performing its competitors in the majority of experiments in terms of our chosen metrics. In Table B.1 in Appendix B we report the proportion of times the spline-based input model attained the smallest maximum gap and smallest integrated absolute difference over the $G = 500$ fits of the arrival rate function for all 27 experiments. In all cases the proportion of times the spline does better than its competitors in terms of the two metrics is over a half, and in many cases this proportion is equal to, or very close to, 1.

Reflecting on the experiment as a whole, as the number of sets of observations, m_d, increases all methods improved for both metrics. Given more data the optimal number of intervals, set using the Chen and Schmeiser (2018) pre-processing method, increases allowing the piecewise-linear and piecewise-quadratic methods to attain a better fit as seen from Figure 5.5.1 to Figure 5.5.2. Another observation made was that for fixed m_d and peak duration ξ, as the peak height, κ, increases both average metrics increase. This indicates that all the methods found abrupt peaks in the arrival rate challenging to estimate. This effect was reflected in Figure 5.5.1 and the location of the maximum absolute difference in the arrival rates; for arrival rate functions with sharp peaks the maximum difference often fell close to the centre of the peak.



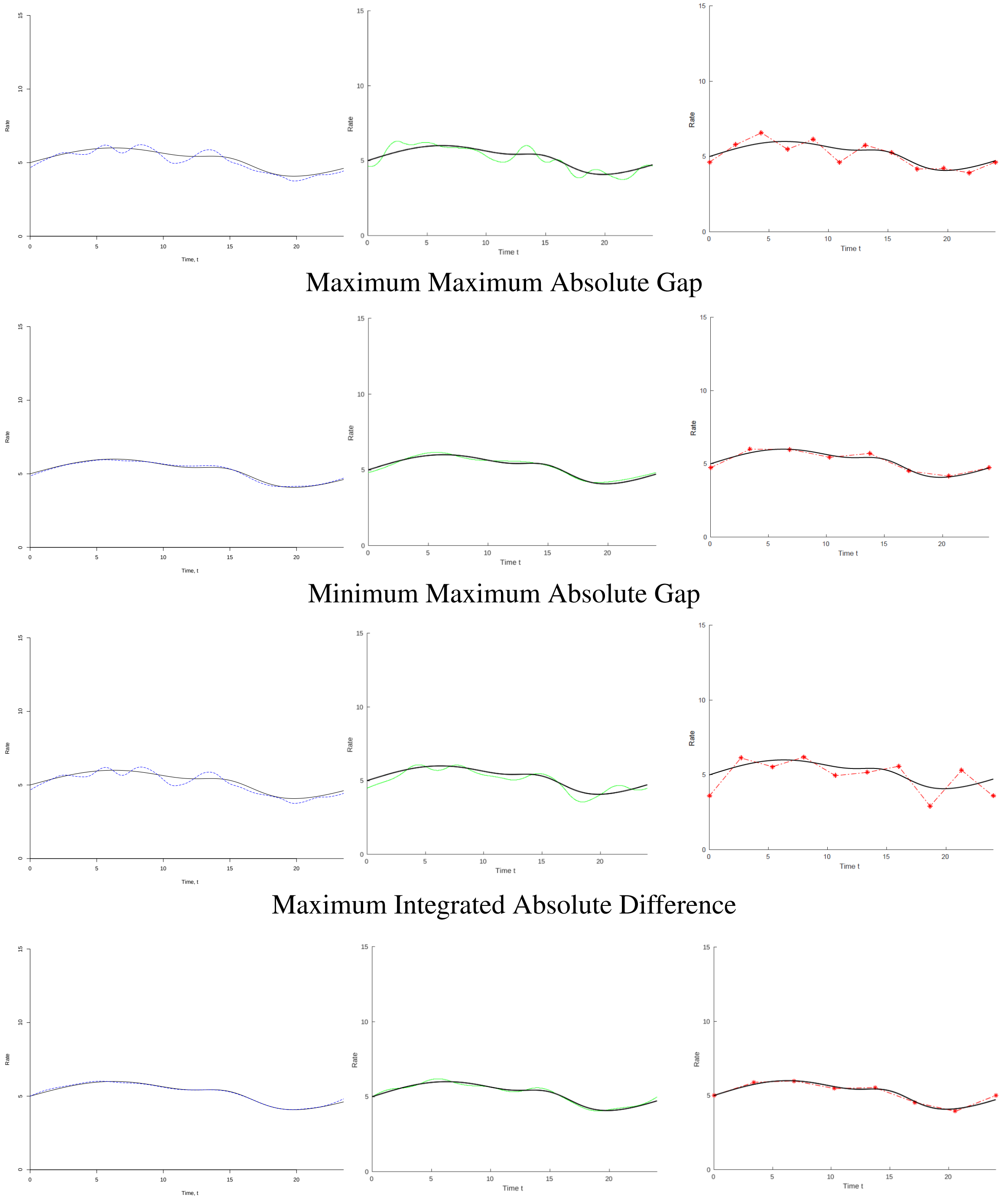
Maximum Maximum Absolute Gap

Minimum Maximum Absolute Gap

Maximum Integrated Absolute Difference

Minimum Integrated Absolute Difference

Figure 5.5.1: $m_d = 15$, $\kappa = 5$, $\xi = 1$ - SPL (blue), PQ (green) and PL (red)



Maximum Maximum Absolute Gap

Minimum Maximum Absolute Gap

Maximum Integrated Absolute Difference

Minimum Integrated Absolute Difference

Figure 5.5.2: $m_d = 100$, $\kappa = 1$, $\xi = 10$ - SPL (blue), PQ (green) and PL (red)

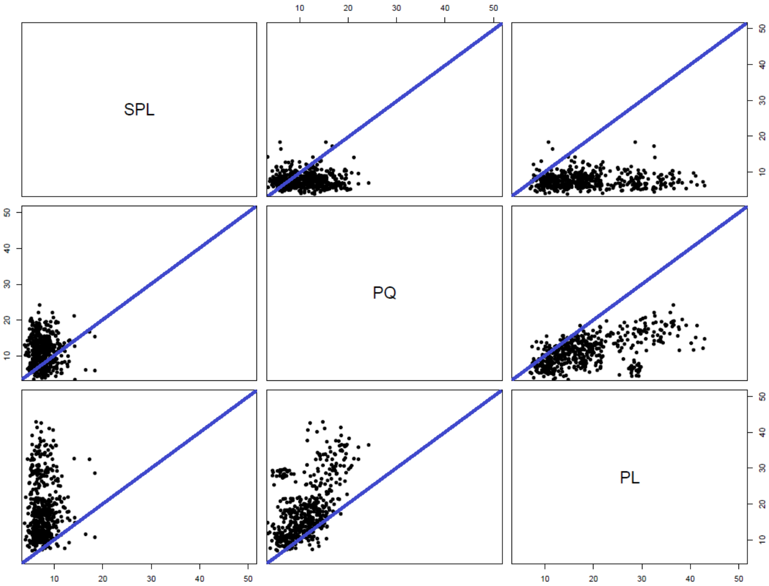


Figure 5.5.3: Pairwise comparison of the three methods using scatter plots of the integrated absolute difference, δ, over $G = 500$ replications of the NHPP fit. Here $m_d = 15$, $\kappa = 5$ and $\xi = 1$.

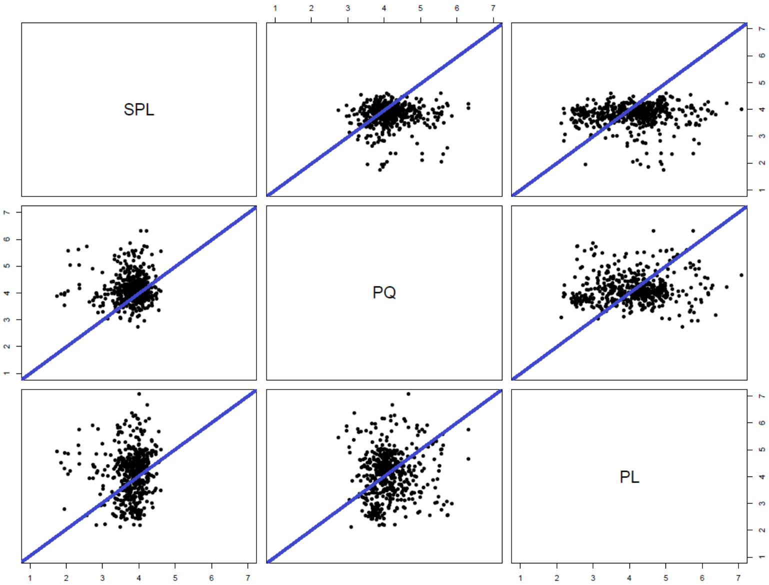


Figure 5.5.4: Pairwise comparison of the three methods using scatter plots of the maximum absolute difference, ζ, over $G = 500$ replications of the NHPP fit. Here $m_d = 15$, $\kappa = 5$ and $\xi = 1$.

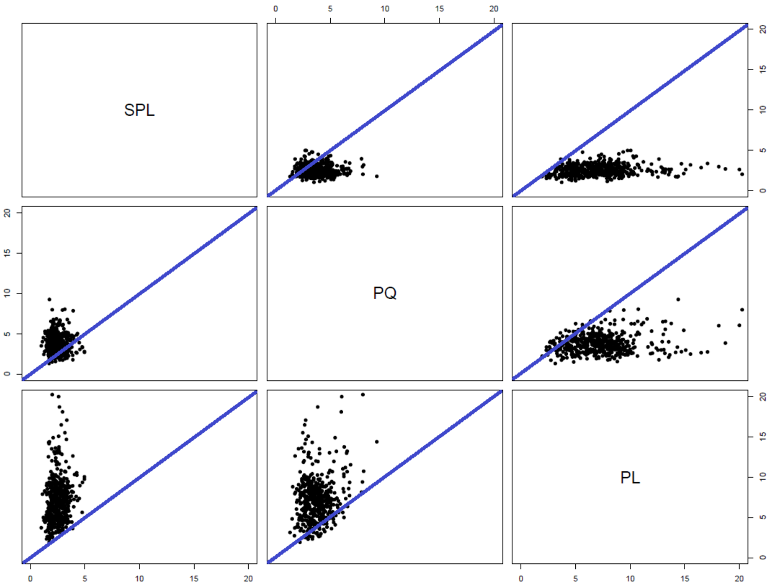


Figure 5.5.5: Pairwise comparison of the three methods using scatter plots of the integrated absolute difference, δ, over $G = 500$ replications of the NHPP fit. Here $m_d = 100$, $\kappa = 1$ and $\xi = 10$.

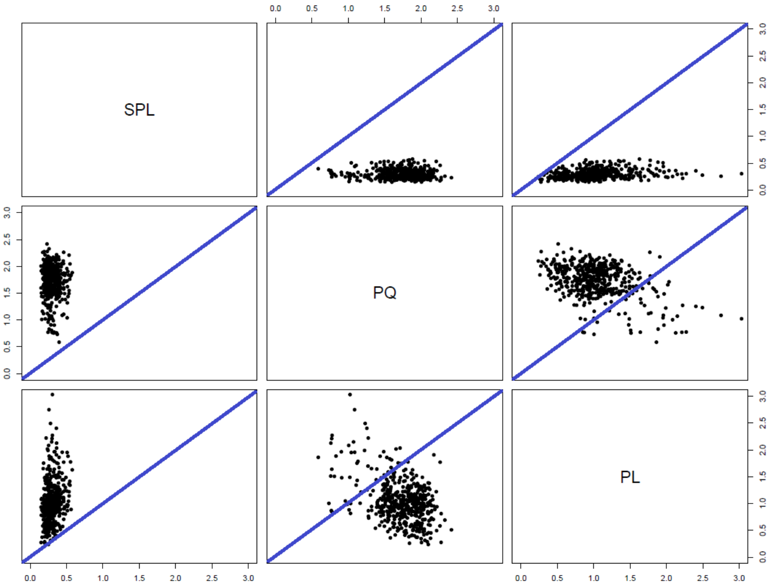


Figure 5.5.6: Pairwise comparison of the three methods using scatter plots of the maximum absolute difference, ζ, over $G = 500$ replications of the NHPP fit. Here $m_d = 100$, $\kappa = 1$ and $\xi = 10$.

5.5.2 Realistic Example

Using the methodology outlined in §5.3 we fit a spline function given arrival-time observations from a real-world A&E department. The arrival rate to A&E is believed to follow a cyclic pattern over a week long period. We focus on observations over the summer months: June, July and August, from the years 2011/12 as we believe the weekly arrival behaviour in this period to be similar. Summer is also the season with the lowest number of public holidays which are believed to cause fluctuations to arrivals to A&E. We therefore have $m_d = 24$ observed weeks of the A&E department.

Before fitting the spline function we considered the assumption that the arrivals follow a NHPP. Using a chi-square goodness-of-fit test we checked whether the total number of arrivals on each day of the week could be said to be Poisson. In conclusion we had significant evidence to reject that the arrival counts on Wednesdays, Thursdays and Fridays were Poisson; the counts on these days were particularly overdispersed in comparison to a Poisson distribution. Despite this, in reality, NHPPs are often used as input models without checking such assumptions. We will therefore proceed to fit the arrival data using our spline-based method but we will also fit the arrival rate using the MNO-PQRS method as it has no dependency on the input process being Poisson. The spline fit, constructed from 56 uniformly spaced knot points, can be seen in Figure 5.5.7 along with a 95% pointwise confidence interval. The choice of 56 knots corresponds to a knot every 3 hours with knots at the same time each day; note that, although placement of the knots at the same time is not necessary, it seemed natural in this cyclic context. The MNO-PQRS fit can be seen in Figure 5.5.8. Note that prior to running MNO-PQRS the pre-processing method of Chen and Schmeiser (2018) was used and split the week into 88 intervals of equal length.

The resulting representations in Figures 5.5.7 and 5.5.8 exhibit very similar behaviour. On Thursday and Friday, the 4^{th} and 5^{th} cycles in the arrival rate, where the p-value of the goodness-of-fit test was particularly significant ($< 1 \times 10^{-6}$) we appear to see the most discrepancy between the fits but even there the difference is not great. Of course, since this

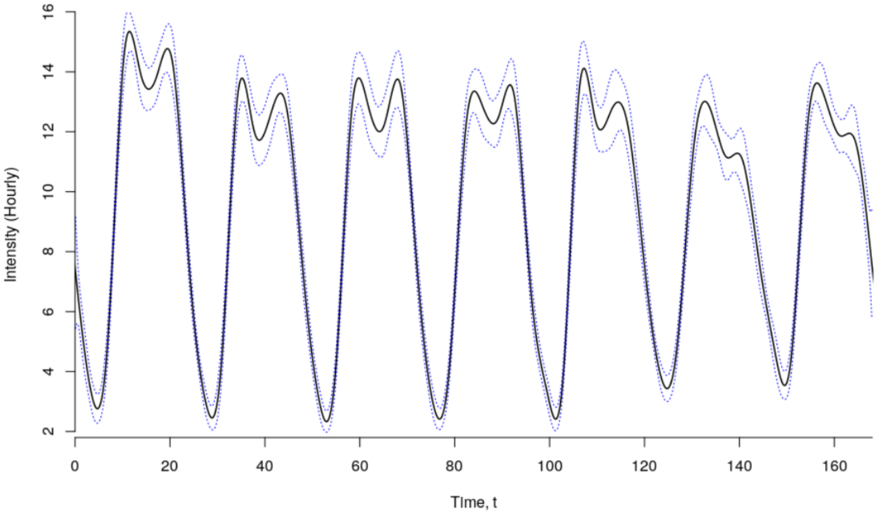


Figure 5.5.7: Spline-based fit to A&E weekly observations with a 95% pointwise confidence interval.

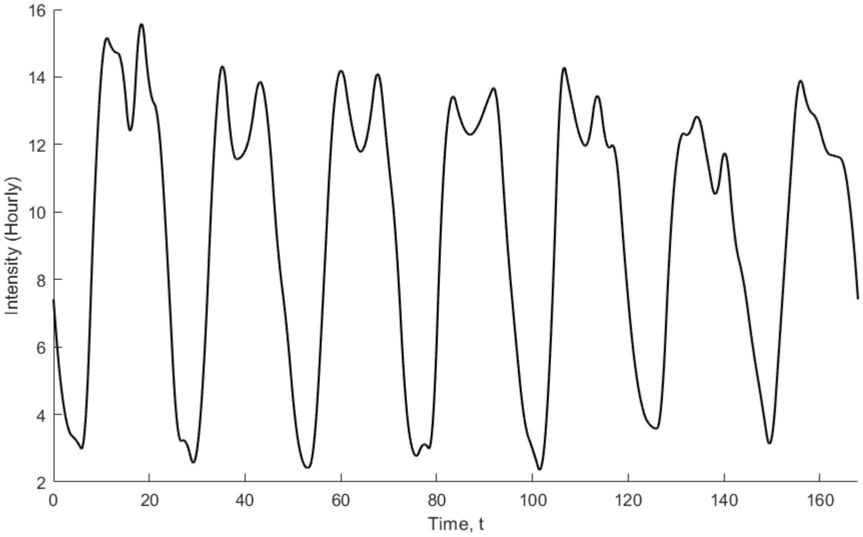


Figure 5.5.8: MNO-PQRS fit to A&E weekly observations.

is real-world example we do not know the true underlying arrival rate, and therefore which method is closer to the truth, but the similarity between the representations indicates that the spline method is not greatly sensitive to data which diverges from Poisson assumptions.

5.5.3 Under- and Overdispersed Data

Motivated by the real-world experiment in §5.5.2, we tested the robustness of the methods to non-Poisson observations using simulated data from nonhomogeneous non-Poisson processes with arrival rate functions as described in the controlled experiment in §5.5.1. Specifically, we considered both underdispersed and overdispersed data. Oreshkin et al. (2016) discuss modelling the arrival rate in call centre systems where the variance of daily arrival counts is typically larger than the mean, causing overdispersed arrivals. Sellers and Morris (2017) discuss causes of underdispersion in data and possible models to account for it. To test the robustness of the methods to under- and overdispersed data, arrival times were generated from a Markov-MECO process using the Markov-MECO-based tool for generating nonhomogeneous non-renewal arrival processes presented by Nelson and Gerhardt (2011). Note that, we are interested in the robustness of the method to non-Poisson data and the tool presented by Nelson and Gerhardt (2011) also allows incorporation of correlation between arrivals. As this is a separate issue, possibly for future consideration, we set correlation in the Markov-MECO process to 0. The Markov-MECO-based tool allows the user to select a target squared coefficient of variation, cv^2, of the process. For Poisson distributed data the squared coefficient of variation, cv^2, equals 1, by definition of a Poisson process. In this experiment we consider both underdispersed, $cv^2 = 0.5$, and overdispersed, $cv^2 = 1.5$ data; the averaged metrics from fitting $G = 500$ rate functions given under- and overdispersed data are presented in Tables 5.5.2 and 5.5.3 respectively.

In Tables 5.5.2 and 5.5.3 we see that the spline-based method gives the smallest average integrated absolute difference and smallest average maximum gap in all experiments. This held for all 27 arrival rate functions using both under- and overdispersed data. In Tables B.2 and B.3 in Appendix B we report the proportion of times the spline-based input model attained the smallest maximum gap and smallest integrated absolute difference over the $G = 500$ fits of the arrival rate function for all 27 experiments given under- and overdispersed observations. The results reflect those from our first experiment, in §5.5.1, where the observations were Poisson; in all cases the proportion of times the spline does better than its competitors in terms of the two metrics is over a half, and in many cases this proportion is equal to, or very close to, 1.

By considering the same arrival rate functions presented in §5.5.1 we can directly compare the results of the fit of the arrival rate function for underdispersed, Poisson and overdispersed data. From results Tables 5.5.1, 5.5.2 and 5.5.3, it is clear that all methods do worse when the arrivals are overdispersed; this held for all experiments. It also appears that the spline-based method performed similarly for underdispersed and Poisson arrivals for all levels of data, m_d. Note that, although the coefficient of variation, ι, remained small for all experiments, the spline-based function no longer consistently achieved the smallest value given under- and overdispersed data.

Within the parameters of this experiment we have demonstrated that when the arrival data departs from Poisson, by being over- or underdispersed, the spline-based method still performs the best in terms of the metrics of average integrated absolute difference and

Table 5.5.2: The average maximum absolute difference, $\bar{\zeta}$, the average integrated absolute difference, $\bar{\delta}$, and the coefficient of variation of the integrated absolute difference, ι, for the fit of two arrival rate functions given different settings of m_d, κ and ξ for underdispersed data.

cv^2=0.5	$m_d = 15$, $\kappa = 5$, $\xi = 1$			$m_d = 100$, $\kappa = 1$, $\xi = 10$		
	$\bar{\zeta}$ (se)	$\bar{\delta}$ (se)	ι	$\bar{\zeta}$ (se)	$\bar{\delta}$ (se)	ι
SPL	3.42 (3.27×10^{-2})	7.37 (7.81×10^{-2})	0.24	0.34 (4.75×10^{-3})	2.97 (5.38×10^{-2})	0.19
PQ	4.08 (1.09×10^{-2})	7.60 (9.40×10^{-2})	0.28	1.78 (9.58×10^{-3})	3.34 (5.03×10^{-2})	0.20
PL	4.25 (2.94×10^{-2})	13.66 (2.61×10^{-1})	0.43	0.89 (1.29×10^{-2})	6.50 (1.03×10^{-1})	0.35

Table 5.5.3: The average maximum absolute difference, $\bar{\zeta}$, the average integrated absolute difference, $\bar{\delta}$, and the coefficient of variation of the integrated absolute difference, ι, for the fit of two arrival rate functions given different settings of m_d, κ and ξ for overdispersed data.

cv^2=1.5	$m_d = 15$, $\kappa = 5$, $\xi = 1$			$m_d = 100$, $\kappa = 1$, $\xi = 10$		
	$\bar{\zeta}$ (se)	$\bar{\delta}$ (se)	ι	$\bar{\zeta}$ (se)	$\bar{\delta}$ (se)	ι
SPL	3.74 (2.64×10^{-2})	10.05 (1.52×10^{-1})	0.34	0.47 (7.50×10^{-3})	5.21 (1.25×10^{-1})	0.53
PQ	4.26 (3.32×10^{-2})	15.65 (2.16×10^{-1})	0.31	1.71 (1.56×10^{-3})	6.62 (1.19×10^{-1})	0.40
PL	4.86 (5.89×10^{-2})	27.15 (4.59×10^{-1})	0.37	1.77 (4.37×10^{-2})	11.52 (2.47×10^{-1})	0.48

average maximum absolute gap compared to the piecewise-quadratic and piecewise-linear methods. This indicates that the spline-based input modelling approach is robust to arrivals that are under- or overdispersed in comparison to a Poisson distribution.

5.6 Conclusion

We have provided a spline function input modelling method based on the penalised log-likelihood for fitting the rate function of a NHPP given arrival-time observations. In comparison to two recent methods in the literature, the spline-based method was seen to perform best in terms of the average integrated absolute difference and average maximum absolute gap in all experiments, including experiments in which the provided observations were non-Poisson. The chosen metrics are indicators of how well a method recovers the true arrival rate function of a NHPP. We have therefore presented a spline-based input modelling method that has been shown in our experiments to consistently recover the arrival rate function of a NHPP better than appropriate competitors in the literature and is robust to under- and overdispersed observations.

A realistic input modelling situation, given observations from an A&E department, was also presented. We showed that, even when the observations departed from Poisson assumptions, the spline-based technique returned a similar rate function to the one produced by MNO-PQRS, which was designed to fit the rate functions of general input processes. Although not the main topic of this chapter in practice given real-world data we could test the quality of the fit of our spline-based model by splitting our data into a test and training set. Fitting the spline-based model using the training set would then allow us to check the model fit using real-world data from the test set.

We also presented a simple thinning-based method for simulating arrivals from the resulting spline arrival rate function. This took advantage of the composition of the spline function as a linear combination of B-spline basis functions with known maximums.

In practice, arrival counts are sometimes recorded instead of arrival times. This method could be extended to work for arrival count data through a simple modification of the log-likelihood. In the same way, provided an appropriate likelihood can be derived, the penalise log-likelihood method could be extended for use with other non-stationary non-Poisson arrival processes. We leave these extensions for future work.

Another area of further work would be a study of how to choose a suitably large number of knots from which to build the spline function. The spline-based model will lack flexibility if the number of knots is too small, but too many knots can lead to a discrepancy between the second-order approximation used in the trust region algorithm and the objective function. Choosing a “large enough” number of knots is therefore a question of interest.

Conclusions

In this chapter the thesis is concluded by reflecting on the contributions made to the areas of input modelling and input modelling error quantification. Proposals of how the methodology might be extended are also presented.

6.1 Summary of Contributions

In this section a summary of the main findings of this thesis are presented. Contributions have been made to the areas of input modelling and quantification of error caused by input modelling in simulation. A particular focus was on the development of new methodology for the quantification of input modelling errors in simulation models with nonhomogenous input models, specifically nonhomogenous Poisson arrival processes (NHPPs). As previously discussed, ignoring error caused by input modelling can lead to over-confidence in the output of a simulation and therefore the decisions made using it. The methods presented in this thesis are therefore beneficial in practice due to our focus on quantifying and reducing the error caused by input modelling in simulation.

The first contribution of this thesis, presented in Chapter 3, was the development of two techniques for the quantification of input uncertainty in simulation models with piecewise-constant non-stationary Poisson arrival processes. These methods are the first to tackle input uncertainty quantification for simulation models with nonhomogeneous arrival processes. This is a natural step to take in the input uncertainty quantification literature as many systems exhibit nonhomogeneous behaviour in the real-world. This being a first step the methods we produced still had some disadvantages. For instance it is known that, in reality, it is unrealistic to assume that the rate of a NHPP changes instantaneously in time; modelling the arrival process using a piecewise-constant function is therefore something that could be improved upon. That said, the piecewise-constant representation was key to the extension of the methods by Cheng and Holland (1997) and Song and Nelson (2013) for simulation models with time homogeneous input models. Piecewise-constant arrival processes are also often used by practitioners due to their flexibility and ease to use/ understand, this methodology therefore has a good chance of being translated into practice.

In Chapter 4 multiple contributions were presented. The key contribution was a bias detection test with controlled power for detecting bias of a concerning size. Within this the first approach to quantifying bias caused by input modelling was provided, this, again for the first time, allowed a summary of the mean squared error caused by input modelling to be made. Previously bias caused by input modelling had been virtually ignored in the input modelling error literature. The proposed approach to tackling bias caused by input modelling was not to aim straight at gaining an accurate estimate of bias, instead it tested whether bias was relevant. The bias detection test presented in Chapter 4 tests for non-zero bias. Developing the bias detection test in this way is advantageous as when bias is small, and therefore not of much interest, detecting that is not significantly different to zero is computationally much cheaper than trying to accurately estimate it. Also, by controlling the power of the test, there is a high probability of rejecting the null hypothesis when bias is higher than a threshold value deemed by the practitioner to be the smallest value of bias of concern to them. One downside of the method is its scalability to the number of input models. Simulation models often have many inputs so this is an issue of some concern. To try and tackle this issue the method of Sanchez and Sanchez (2005) for Resolution V experimental designs was utilised to reduce the number of design points used to fit the response surface model within the bias detection test. The result of using this method was promising; for a fraction of the computational effort the same conclusions were reached from the bias detection test using a greatly reduced number of design points.

The final contribution of the thesis was a spline-based arrival process modelling method for representing the arrival rate function of a NHPP. The spline-based method, presented in Chapter 5, also led to a simple way to generate arrival observations for use within a simulation experiment. The aim of developing the spline-based model was to create an input modelling method that could recover the underlying arrival rate of a NHPP better than its competitors; in doing so it would pass less input modelling error to the output of the simulation. The spline-based method achieved promising results. In a controlled experiment, compared to two recent methods in the literature, a piecewise-linear approach and a piecewise-quadratic approach, the spline-based model was shown to attain lower integrated absolute difference and maximum absolute difference on average. The spline-based model was also shown to have more stability than its piecewise-linear and piecewise-quadratic competitors. Another consideration was the robustness of the spline-based method to over- or underdispersed data. By repeating the controlled experiment with both over- and underdispersed observations, it was found that the spline-based method performed better than its competitors even though the piecewise-quadratic model made no assumption that the input process had to be Poisson. In practice, due to the flexibility of spline functions, the spline-based method could be used to model a wide variety of arrival rate functions. In the next section possible future steps as discussed for the spline-based input model, amongst other possible research directions.

6.2 Further Work

In this section possible extensions to the methodology described in this thesis and ideas of how these extensions might be achieved are presented. First a comparison of the input modelling techniques discussed in Chapter 5 in terms of the variability they pass to the output of a simulation is proposed. It is believed this will show that on average the spline-based input modelling method passes less error to the output of a simulation. Further work in input uncertainty quantification for simulation models with smoother than piecewise-constant input processes is also considered.

6.2.1 A Controlled Comparison of Input Modelling Methods in Terms of the Error Passed to the Simulation Response

In Chapter 5 the spline-based input model was shown to perform better than its competitors in terms of recovering the true arrival rate function of a NHPP. A natural extension to this would therefore be to show that, in comparison to other input modelling techniques, the spline-based input model also propagates less input modelling error to the output of a simulation model.

As a direct follow up to the controlled experiment in Chapter 5 a comparison of the spline-based method and the methods of Chen and Schmeiser (2017) and Zheng and Glynn (2017) in terms of the input modelling error passed to the output of interest of a simulation is proposed. This experiment is designed to show, in a controlled experiment where all other factors, such as arrival rate, observed data and random seed, are kept constant, that the variability in the simulation response is smaller when the spline-based input model is used. Using the same seed and observed data in the comparison means all three input modelling methods are treated equally; this allows any difference in the variation of the methods to be attributed the input modelling technique.

Given a NHPP with known arrival rate function, $\lambda^c(t)$, any number of sets of observations can be generated from the true arrival process. Let an observation of the arrival process, $\boldsymbol{X}_j$, correspond to all of the arrivals in that time unit, where for example a_j arrivals occur on the j^{th} day of observation, $\boldsymbol{X}_j = \{X_{j1}, X_{j2}, \ldots, X_{ja_j}\}$. A set of m observations is therefore denoted $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m\}$. For a set of m observations each input modelling technique can be used to fit an estimate arrival rate, $\widehat{\lambda}(t)$. Let the spline-based input model be denoted $\widehat{\lambda}_{\text{SPL}}(t)$, the piecewise-quadratic input model of Chen and Schmeiser (2017)

$\widehat{\lambda}_{\mathrm{PQ}}(t)$, and the piecewise-linear model of Zheng and Glynn (2017) $\widehat{\lambda}_{\mathrm{PL}}(t)$. Repeating this process many, say G, times allows many arrival rate representations to be fitted; in Chapter 5 at this point the input modelling methods were compared by considering metrics that indicated how well the input modelling methods recovered the true arrival rate function, $\lambda^c(t)$. To take the next step and consider error caused by input modelling in simulation, the true arrival rate and each fit of the arrival rate would be used to drive a large number of replications, r, of a given simulation model. In replicating each simulation a large number of times times the stochastic estimation error in the simulation response is driven down.

The experiment to compare the amount of error passed to the simulation response from the three NHPP input modelling techniques would proceed as follows:

1. Given a NHPP with true arrival rate function $\lambda^c(t)$
2. For i from 1 to G
    (a) Simulate m observations, $\{\boldsymbol{X}_{i1}, \boldsymbol{X}_{i2}, \ldots, \boldsymbol{X}_{im}\}$, from the true arrival rate function, $\lambda^c(t)$.
    (b) Given $\{\boldsymbol{X}_{i1}, \boldsymbol{X}_{i2}, \ldots, \boldsymbol{X}_{im}\}$, fit the NHPP arrival rate function using:
        i. the spline-based model to get the estimate $\widehat{\lambda}^i_{\mathrm{SPL}}(t)$,
        ii. the piecewise-quadratic model to get the estimate $\widehat{\lambda}^i_{\mathrm{PQ}}(t)$,
        iii. and the piecewise-linear model to get the estimate $\widehat{\lambda}^i_{\mathrm{PL}}(t)$.
    (c) Given the true input model and each fitted input model, starting from the same seed, run r replications of the simulation to attain:
        i. $Y_j(\lambda^{c^i}(t))$, for $j = 1, 2, \ldots, r$,
        ii. $Y_j(\widehat{\lambda}^i_{\mathrm{SPL}}(t))$, for $j = 1, 2, \ldots, r$,
        iii. $Y_j(\widehat{\lambda}^i_{\mathrm{PQ}}(t))$, for $j = 1, 2, \ldots, r$,
        iv. $Y_j(\widehat{\lambda}^i_{\mathrm{PL}}(t))$, for $j = 1, 2, \ldots, r$.

    where $Y_j(\cdot)$ denotes the output of the j^{th} simulation replication.

(d) Calculate the average simulation response over the r replications, $\bar{Y}(\lambda^{ci}(t))$, $\bar{Y}(\widehat{\lambda}^{i}_{\mathrm{SPL}}(t))$, $\bar{Y}(\widehat{\lambda}^{i}_{\mathrm{PQ}}(t))$ and $\bar{Y}(\widehat{\lambda}^{i}_{\mathrm{PL}}(t))$.

3. Given the average simulation responses from the three methods using G sets of input data, report the variability of the average simulation response for each input modelling method. Also report the difference between the average simulation response for the true input model, i.e the simulation response when there is no input modelling error, and the average simulation response for each of the input modelling methods.

Comparing the variability in the average simulation response, $\bar{Y}(\widehat{\lambda}(t))$, given a large number of sets of m observations will highlight which input modelling technique passes the most input uncertainty to the simulation response. Also, comparing the difference between the simulation response driven by the true input model, $\lambda^{c}(t)$, and the fitted input models, $\widehat{\lambda}_{\mathrm{SPL}}(t), \widehat{\lambda}_{\mathrm{PQ}}(t)$ and $\widehat{\lambda}_{\mathrm{PL}}(t)$, will highlight which input modelling technique passes the most bias caused by input modelling to the simulation response.

In this experiment, interest is in the effect of the choice of input modelling technique on the amount of IU passed to the output. Consideration of the amount of input data, the shape of the arrival rate function and the shape of the simulation response surface given a fixed input modelling method may also be of interest to consider. In Chapter 5 the amount of input data and the shape of the arrival rate function were considered and on average the spline-based input model was able to recover the true NHPP arrival rate better than its competitors in terms of integrated absolute difference and maximum absolute difference; this is a promising indication that the error passed to the output of the simulation will also be lower. The experiment proposed in this section will allow a conclusion of whether this is the case.

6.2.2 Quantifying Input Uncertainty for Smoother than Piecewise-Constant Input Models

In this thesis new methodology for input uncertainty quantification in simulation models with inputs having piecewise-constant nonhomogeneous arrival processes was provided in Chapter 3. Input uncertainty quantification for simulation models with smoother (than piecewise-constant) arrival rate functions was not considered. This is partly due to the lack of a definition for input uncertainty in this context. The extension to NHPPs with piecewise-constant rate functions meant input uncertainty could easily be broken down; the total input uncertainty of the arrival process was equal to the contribution of each independent interval of the arrival process. This has its advantages as it allows a practitioner to see the intervals in which follow up data collection would be needed most according to which interval was contributing most to the overall input uncertainty. For smoother than piecewise-constant functions the definition of IU is a little harder to determine. Deciding whether IU should be defined at a point in time or over an interval is a question of interest and may depend on the simulation performance measure under study.

A natural step forward following this thesis would be to develop methodology for the quantification of input modelling error in simulation models with smoother (than piecewise-constant) arrival processes. Throughout this thesis the importance of taking input modelling error into account in the summary of error in the simulation response has been argued repeatedly. There is therefore a strong motivation to continue research in this area.

One proposed approach to give a quick estimate of the total IU contribution of an arrival process, and that could be used with any nonhomogeneous input model would be to use bootstrapping, see Ankenman and Nelson (2012). The intuition behind bootstrapping is that as the amount of real-world observations increases the estimated arrival rate, $\widehat{\lambda}(t)$, will get closer to the true, unknown, arrival rate function $\lambda^c(t)$. Generating new samples using $\widehat{\lambda}(t)$ mimics having collected further real-world samples, and these samples become more like samples from $\lambda^c(t)$ as the number of real-world observations available to estimate $\widehat{\lambda}(t)$

increases. Quantifying IU using bootstrapping would first mean fitting an estimate arrival rate, $\widehat{\lambda}(t)$, to the observed real-world data, $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m\}$, then, treating $\widehat{\lambda}(t)$ as if it were the true arrival rate, and applying bootstrapping to generate further samples of data. Let us denote the i^{th} set of bootstrap observations by $\{\boldsymbol{X}^\star_{i1}, \boldsymbol{X}^\star_{i2}, \ldots, \boldsymbol{X}^\star_{im}\}$. Given observations $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m\}$ from the true arrival process, input uncertainty can be estimated using bootstrapping as follows:

1. Given real-world observations $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m\}$, fit the estimate arrival rate function, $\widehat{\lambda}(t)$.

2. For i from 1 to b:

   (a) Given $\widehat{\lambda}(t)$, simulate m bootstrap observations, $\{\boldsymbol{X}^\star_{i1}, \boldsymbol{X}^\star_{i2}, \ldots, \boldsymbol{X}^\star_{im}\}$.

   (b) Using bootstrap observations $\{\boldsymbol{X}^\star_{i1}, \boldsymbol{X}^\star_{i2}, \ldots, \boldsymbol{X}^\star_{im}\}$, fit the bootstrap input model $\widehat{\lambda}^\star_i(t)$.

   (c) Complete r replications of the simulation $Y_j(\widehat{\lambda}^\star_i(t))$, $j = 1, 2, \ldots, r$, driven using bootstrap input model $\widehat{\lambda}^\star_i(t)$, to attain $\bar{Y}(\widehat{\lambda}^\star_i(t))$.

3. Estimate the input uncertainty in the output of the simulation.

In the final step input uncertainty in the simulation response would be estimated using the same approach as Ankenman and Nelson (2012). They proposed to approximate input uncertainty, $\widehat{\sigma}^2_I$, by estimating the total simulation variance, $\widehat{\sigma}^2_T$, and then removing from it the simulation estimation error $\widehat{\sigma}^2_S$ as follows

$$\widehat{\sigma}^2_I = \frac{\widehat{\sigma}^2_T - \widehat{\sigma}^2_S}{r},$$

where

$$\widehat{\sigma}^2_T = \frac{r}{(b-1)} \sum_{i=1}^{b} (\bar{Y}_{i.} - \bar{Y}_{..})^2,$$

$$\widehat{\sigma}^2_S = \frac{1}{b(r-1)} \sum_{i=1}^{b} \sum_{j=1}^{r} (Y_{ij} - \bar{Y}_{i.})^2.$$

Here '.' denotes averaging over an index, for example $\bar{Y}_{i.}$ means averaging over the r simulation replications, $Y_j(\widehat{\lambda}_i^{\star}(t))$ for $j = 1, 2, \ldots, r$. The intuition behind this approximation is that the total simulation variance, σ_T^2, measures both stochastic estimation error and input uncertainty and thus by removing the stochastic estimation error an estimate of the input uncertainty in the simulation response remains.

This method will allow a practitioner to estimate the input uncertainty contributions from any input processes used to drive the simulation, but it does have some problems. One drawback is that bootstrap estimators can be inaccurate, for example when input uncertainty is small, it is possible to return a negative estimate of it using this approach since $\widehat{\sigma}_T^2$ and $\widehat{\sigma}_S^2$ are both estimates. There is also the issue that bootstrapping can be computationally expensive. To get a good approximation of IU many bootstrap samples may be required, and each bootstrap input model may require a considerable number of replications of the simulation to drive down the stochastic estimation error.

Another approach to quantifying IU in simulation models with nonhomogeneous input processes would be to focus on nonhomogeneous Poisson processes and work towards quantifying the input modelling error passed to the simulation output given a certain input modelling technique. In Chapter 5 a spline-based input model for the arrival rate function of a NHPP was proposed. The bootstrap approach outlined above could be used to estimate the input uncertainty passed from the spline-based model to the simulation response, but, given the drawbacks of the bootstrap approach, an open question remains as to whether new methodology could be developed to quantify the input uncertainty contribution of this input model.

Recall that in the piecewise-constant case, to avoid the need for bootstrapping, asymptotic approximations of the MLE distributions were used. Within the spline-based input modelling method n spline coefficients, $\boldsymbol{c} = \{c_1, c_2, \ldots, c_n\}$, control the shape of the resulting arrival rate model. These spline coefficients are estimated by maximising the penalised log-likelihood of a NHPP for a fixed penalty parameter, θ. Recall that the penalty parameter is selected by finding the combination $\{\widehat{\boldsymbol{c}}_{\theta}, \theta\}$ that attains the minimum RIC score.

The estimation of the spline coefficients is therefore directly affected by the observed data and the choice of θ. Thus error in either the spline coefficients or the penalty will pass input modelling error to the simulation response. There are therefore $n+1$ input parameters to consider in this problem. Note that there already exist approximations for the variance of the spline coefficients, $\widehat{\boldsymbol{c}}_\theta$, in the context of penalised likelihood estimation, but these approximations do not take into account the error in the choice of the penalty parameter, θ.

The main complication in quantifying the input uncertainty contribution of the spline-based input model is how to consider the penalty parameter, θ. One option would be to fix the penalty. This would reduce quantifying the input uncertainty contribution of the arrival process to considering the error passed to the simulation response from the estimation of the spline coefficients. But this approach ignores the possible variability in θ over different fits of the arrival rate function. Note that many input uncertainty quantification techniques, including those of Cheng and Holland (1998) and Song and Nelson (2015) as discussed in Chapter 3, require repeatedly re-estimating the parameters of the input model. For the spline-based model this would mean maximising the penalised log-likelihood for the given fixed penalty many times.

In Chapter 5 a controlled experiment was conducted in which the spline-based model was used to fit $G = 500$ arrival rate functions given 500 sets of m days of observations, where all observations were simulated from the same, known, arrival rate function. In this experiment the penalty was not fixed; for each fit of the arrival rate the penalty was chosen by finding the combination $\{\widehat{\boldsymbol{c}}_\theta, \theta\}$ that minimised the RIC score. This resulted in a considerable amount of variability in the value of the penalty, although, as m was increased the penalty could be seen to become more stable as it was pushed down to 0. The variability observed in θ in this controlled experiment indicates that by fixing θ in the consideration of input uncertainty, the combination $\{\widehat{\boldsymbol{c}}_\theta, \theta\}$ may not attain an RIC score close to the minimum. This could lead to under- or over smoothing of the arrival rate model which may impact the input uncertainty passed to the simulation response. The key here is to identify whether the variability of the spline-based model for a fixed penalty, $\mathrm{Cov}[\,\widehat{\boldsymbol{c}}_\theta]$, is much smaller than the variability of the spline-based model when θ is re-estimated for each fit, $\mathrm{Cov}[\,\widehat{\boldsymbol{c}}_{\widehat{\theta}}]$. If this is the case ignoring the error in the penalty parameter will lead to underestimating the input uncertainty passed to the simulation response. This requires further investigation.

In this chapter we summarised the main contributions of the thesis and presented, with some detail, future research directions that may be of interest to us going forward.

Appendix

A Proof of Results - Detecting Bias due to Input Modelling in Computer Simulation

A.1 Variability of the Jackknife Estimator of Bias

The jackknife method is an alternative to the delta method that can be used for bias estimation. Usually when quantifying bias we refer to the bias of a statistic of interest, for example a population parameter given a sample of data; in this case let us denote the jackknife estimator of bias $\widehat{b}_{JK}$. In stochastic simulation the statistic we would like to examine is the expected value of the simulation response, $\eta(\boldsymbol{\theta}^{mle})$. However, we can only observe this in the presence of simulation noise. In this appendix we investigate the effect of simulation noise on the variability of the jackknife estimator of bias.

Consider a stochastic simulation model with a single input parameter, θ^c from a single input model. Let θ^{mle} be the maximum likelihood estimator (MLE) of θ^c based on m observations of the input distribution and $\theta^{mle}_{(i)}$ is the "reduced information" MLE based on all but the i^{th} observation. The desired jackknife estimate of bias is

$$\widehat{b}_{JK} = (m-1)\left[\frac{1}{m}\sum_{i=1}^{m}\eta(\boldsymbol{\theta}^{mle}_{(i)}) - \eta(\boldsymbol{\theta}^{mle})\right].$$

Since we cannot evaluate $\eta(\cdot)$ directly, the natural extension to simulation output is,

$$\widehat{b}_{JK+noise} =(m-1)\left[\frac{1}{m}\sum_{i=1}^{m}\frac{1}{r}\sum_{j=1}^{r}Y_j(\theta^{mle}_{(i)}) - \frac{1}{r}\sum_{k=1}^{r}Y_k(\theta^{mle})\right] \tag{A.1}$$

which requires r independent replications of the simulation at each reduced information MLE, $\theta^{mle}_{(i)}$, and, independent of this, r replications of the simulation at the MLE, θ^{mle}. Within (A.1) the output of a replication of the simulation can be decomposed into the expected simulation response plus simulation noise

$$\begin{aligned}\widehat{b}_{JK+noise} =&(m-1)\left[\frac{1}{m}\sum_{i=1}^{m}\frac{1}{r}\sum_{j=1}^{r}(\eta(\theta^{mle}_{(i)})+\varepsilon_{ij}) - \frac{1}{r}\sum_{k=1}^{r}(\eta(\theta^{mle})+\varepsilon_k)\right]\\ =&(m-1)\left[\frac{1}{rm}\sum_{j=1}^{r}\sum_{i=1}^{m}\eta(\theta^{mle}_{(i)}) - \frac{1}{r}\sum_{k=1}^{r}\eta(\theta^{mle}) + \frac{1}{rm}\sum_{j=1}^{r}\sum_{i=1}^{m}\varepsilon_{ij} - \frac{1}{r}\sum_{k=1}^{r}\varepsilon_k\right],\end{aligned} \tag{A.2}$$

where $\varepsilon_{ij} \sim$ i.i.d$(0,\sigma_i^2)$ and $\varepsilon_k \sim$ i.i.d $(0,\sigma_k^2)$. Here (A.2) can be thought of as breaking $\widehat{b}_{JK+noise}$ into $\widehat{b}_{JK}$, the jackknife estimator of bias without simulation noise, and $\widehat{b}_{noise}$, the additional variability in the estimator of bias caused by simulation noise.

The key to this investigation is the variance of $\widehat{b}_{noise}$

$$\begin{aligned}
\operatorname{Var}\left(\widehat{b}_{noise}\right) &= \operatorname{Var}\left((m-1)\left[\frac{1}{rm}\sum_{j=1}^{r}\sum_{i=1}^{m}\varepsilon_{ij} \;-\; \frac{1}{r}\sum_{k=1}^{r}\varepsilon_k\right]\right) \\
&= (m-1)^2\left[\frac{1}{r^2m^2}\sum_{j=1}^{r}\sum_{i=1}^{m}\operatorname{Var}\left(\varepsilon_{ij}\right)+\frac{1}{r^2}\sum_{k=1}^{r}\operatorname{Var}\left(\varepsilon_k\right)\right] \\
&= (m-1)^2\left[\frac{\sigma^2}{rm}+\frac{\sigma^2}{r}\right] \\
&= (m-1)^2\frac{(m+1)\sigma^2}{rm} \qquad \text{(A.3)}
\end{aligned}$$

which is, for large m, is approximately equal to $m^2\sigma^2/r$. This says that, in the presence of simulation noise, the number of simulation replications per reduced information MLE, r, required to maintain a constant level of error as m grows is $r = O(m^2)$, and the total number of simulation replications to compute the jackknife with constant error grows as $O(m^3)$. Thus, it is clear that significant simulation effort may be required; otherwise the jackknife estimate of this bias could be obscured by the presence of simulation noise.

A.2 Asymptotics of b and b^{approx}

Using Taylor series we show that, under certain assumptions, as $m \to \infty$ the bias, $b = \mathrm{E}\left[\eta\left(\boldsymbol{\theta}^{mle}\right)\right] - \eta\left(\boldsymbol{\theta}^{c}\right)$, coincides with the delta approximation of bias, b^{approx}.

ASSUMPTION A.1: Let the expected simulation response, $\eta : \mathbb{R}^k \to \mathbb{R}$,

1. Be three times continuously differentiable in a closed ball G centred at $\boldsymbol{\theta}^c$.

2. Have bounded from above, third-order partial derivatives in the closed ball G, there exists some $M > 0$ such that, for all $\boldsymbol{s} \in G$, $\frac{\partial^3\eta(\boldsymbol{s})}{\partial\theta_i\partial\theta_j\partial\theta_p} \leq M$ for $i, j, p = 1, 2, \ldots, k$.

ASSUMPTION A.2: Let the simulation be driven by L independent, parametric input distributions, with $k \geq L$ input parameters. Assume we have m observations for each of the L distributions. Now let $\boldsymbol{\theta}^{mle} \in \mathbb{R}^k$ be the vector of MLEs given the m observations of each input distribution. We assume the MLEs satisfy standard conditions implying that

1. The MLEs converge in mean, $\mathrm{E}\left(\theta_i^{mle} - \theta_i^c\right) \to 0$ as $m \to \infty$ for $i = 1, 2, \ldots, k$.

2. The MLEs are asymptotically normal, $\sqrt{m}(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^{c}) \xrightarrow{\text{D}} \text{MVN}_k(\mathbf{0},\ \text{I}_0(\boldsymbol{\theta}^{c})^{-1}) = \boldsymbol{Z}$.
3. For some $\varepsilon > 0$, $|\theta_i^{mle} - \theta_i^{c}|^{3+\varepsilon}$ are uniformly integrable for all $m \in \mathbb{N}$, and $i = 1, 2, \ldots, k$.

THEOREM A.1: Let Assumptions A.1 and A.2 hold. Then as $m \to \infty$ the scaled bias, mb, and the scaled delta approximation, mb^{approx}, both converge to

$$\frac{1}{2}\text{tr}(\text{I}_0(\boldsymbol{\theta}^{c})^{-1}\text{H}(\boldsymbol{\theta}^{c})).$$

Proof. Convergence of the MLEs implies that for m large enough we will have $\boldsymbol{\theta}^{mle} \in G$. Therefore, under Assumption A.1.1, the expected simulation response at $\boldsymbol{\theta}^{mle} \in G$ can be expanded via a Taylor series as

$$\eta(\boldsymbol{\theta}^{mle}) = \eta(\boldsymbol{\theta}^{c}) + \nabla\eta(\boldsymbol{\theta}^{c})^T(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^{c}) + \frac{1}{2}(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^{c})^T\text{H}(\boldsymbol{\theta}^{c})(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^{c}) + \Upsilon_3(\boldsymbol{\theta}^{mle}), \tag{A.4}$$

where $\Upsilon_3(\boldsymbol{\theta}^{mle})$ is the remainder, made up of higher-order terms of the Taylor series. For $k \geq 3$ there exists $\boldsymbol{\rho} \in G$ such that

$$\begin{aligned}\Upsilon_3(\boldsymbol{\theta}^{mle}) = \frac{1}{6}\sum_{i=1}^{k}(\theta_i^{mle} - \theta_i^{c})^3\frac{\partial^3\eta(\boldsymbol{\rho})}{\partial\theta_i^{3}} + \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1, j\neq i}^{k}(\theta_i^{mle} - \theta_i^{c})^2(\theta_j^{mle} - \theta_j^{c})\frac{\partial^3\eta(\boldsymbol{\rho})}{\partial\theta_i{}^2\partial\theta_j}\\ + \frac{1}{6}\sum_{i=1}^{k}\sum_{j=1, j\neq i}^{k}\sum_{p=1, p\neq i,\ j}^{k}(\theta_i^{mle} - \theta_i^{c})(\theta_j^{mle} - \theta_j^{c})(\theta_p^{mle} - \theta_p^{c})\frac{\partial^3\eta(\boldsymbol{\rho})}{\partial\theta_i\partial\theta_j\partial\theta_p}.\end{aligned}$$

By taking the expectation of (A.4) we may write bias due to input modelling as

$$\begin{aligned}b = \text{E}[\eta(\boldsymbol{\theta}^{mle})] - \eta(\boldsymbol{\theta}^{c}) = \nabla\eta(\boldsymbol{\theta}^{c})^T\text{E}(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^{c}) + \frac{1}{2}\text{E}\left[(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^{c})^T\text{H}(\boldsymbol{\theta}^{c})(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^{c})\right]\\ + \text{E}[\Upsilon_3(\boldsymbol{\theta}^{mle})].\end{aligned}$$

Note that, the delta approximation of bias only takes into account the second-order term in this expansion

$$b^{approx} = \frac{1}{2}\text{E}\left[(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^{c})^T\text{H}(\boldsymbol{\theta}^{c})(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^{c})\right] = \frac{1}{2}\text{tr}(\Omega\text{H}(\boldsymbol{\theta}^{c}))$$

where $\Omega = \text{Var}(\boldsymbol{\theta}^{mle})$ and, under Assumption A.2.2, $\lim_{m\to\infty} m\Omega = \text{I}_0(\boldsymbol{\theta}^{c})^{-1}$ the inverse Fisher information matrix. We can therefore write $b = b^{approx} + c(\boldsymbol{\theta}^{mle})$; that is, the bias due to input modelling is equal to the delta approximation of bias, b^{approx}, plus a function $c(\cdot)$ containing the expectation of the additional terms of the Taylor expansion evaluated at $\boldsymbol{\theta}^{mle}$. Clearly $mb^{approx} \to \mathrm{tr}(I_0(\boldsymbol{\theta}^c)^{-1}\mathrm{H}(\boldsymbol{\theta}^c))/2$; we will show that $mc(\boldsymbol{\theta}^{mle}) \to 0$.

Consider the expectation of the first order term of the Taylor series expansion. By Assumption A.2.1, $\mathrm{E}(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^c) \to 0$ as $m \to \infty$ and therefore $\nabla\eta(\boldsymbol{\theta}^c)\mathrm{E}(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^c) \to 0$ as $m \to \infty$.

Next consider the expectation of the remainder term, $\mathrm{E}[\Upsilon_3(\boldsymbol{\theta}^{mle})]$. Under Assumption A.1.2 the third-order partial derivatives are bounded above at $\boldsymbol{\rho} \in G$ by $M > 0$ for $i, j, p = 1, 2, \ldots, k$. Thus by linearity of expectation we have,

$$\begin{aligned}\mathrm{E}\left[\Upsilon_3(\boldsymbol{\theta}^{mle})\right] \le \frac{1}{6}\sum_{i=1}^{k}\mathrm{E}[(\theta_i^{mle} - \theta_i^c)^3]M + \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1,j\neq i}^{k}\mathrm{E}[(\theta_i^{mle} - \theta_i^c)^2(\theta_j^{mle} - \theta_j^c)]M \\ +\frac{1}{6}\sum_{i=1}^{k}\sum_{j=1,j\neq i}^{k}\sum_{p=1,p\neq i,\, j}^{k}\mathrm{E}[(\theta_i^{mle} - \theta_i^c)(\theta_j^{mle} - \theta_j^c)(\theta_p^{mle} - \theta_p^c)]M.\end{aligned} \tag{A.5}$$

We will now show that $m \times (A.5)$ converges to 0 as $m \to \infty$ and thus, by sandwich rule, the scaled expectation of the remainder, $m\mathrm{E}\left[\Upsilon_3(\boldsymbol{\theta}^{mle})\right]$, converges to 0. Here the behaviour of the RHS of (A.5) depends on the behaviour of $\mathrm{E}[(\theta_i^{mle} - \theta_i^c)(\theta_j^{mle} - \theta_j^c)(\theta_p^{mle} - \theta_p^c)]$ for $i, j, p = 1, 2, \ldots, k$. Taking the modulus of this expectation and applying Holder's inequality, (Hardy et al., 1952), followed by the arithmetic mean - geometric mean inequality (Abramowitz and Stegun, 1964), we have

$$\begin{aligned}&|\mathrm{E}[(\theta_i^{mle} - \theta_i^c)(\theta_j^{mle} - \theta_j^c)(\theta_p^{mle} - \theta_p^c)]| \qquad \text{(A.6)}\\ &\le \mathrm{E}\left[|(\theta_i^{mle} - \theta_i^c)||(\theta_j^{mle} - \theta_j^c)||(\theta_p^{mle} - \theta_p^c)|\right]\\ &= \mathrm{E}\left[\sqrt[3]{|(\theta_i^{mle} - \theta_i^c)|^3|(\theta_j^{mle} - \theta_j^c)|^3|(\theta_p^{mle} - \theta_p^c)|^3}\right]\\ &\le \frac{1}{3}\mathrm{E}[|(\theta_i^{mle} - \theta_i^c)|^3] + \frac{1}{3}\mathrm{E}[|(\theta_j^{mle} - \theta_j^c)|^3] + \frac{1}{3}\mathrm{E}[|(\theta_p^{mle} - \theta_p^c)|^3].\end{aligned} \tag{A.7}$$

By Assumption A.2.2 and A.2.3, $\sqrt{m}\,\mathrm{E}[|(\boldsymbol{\theta}^{mle} - \boldsymbol{\theta}^c)|^3] \to \mathrm{E}[|\boldsymbol{Z}|^3]$; that is, the third absolute moment of the MLE converges to the third absolute moment of the multivariate normally distributed random variable $\boldsymbol{Z}$ (Osius, 1989). Thus,

$$m^{\frac{3}{2}}\mathrm{E}\left[|(\theta_i^{mle}-\theta_i^c)|^3\right] \to \frac{1}{\sqrt{\pi}}\left(2\,I_0(\boldsymbol{\theta}^c)_{ii}^{-1}\right)^{\frac{3}{2}},$$

as $m \to \infty$ for $i = 1,2,\ldots,k$, (Winkelbauer, 2012). Here $\mathrm{I}_0(\boldsymbol{\theta}^c)_{ii}^{-1}$ is the i^{th} diagonal element of the Fisher information matrix of the joint distribution of the k input parameters. This says that as $m \to \infty$, $m\mathrm{E}\left[|(\theta_i^{mle}-\theta_i^c)|^3\right] \to 0$ for $i = 1,2,\ldots,k$ and therefore $m \times (A.7)$ converges to 0 as well.

By applying the sandwich rule we have $m|\mathrm{E}\left[(\theta_i^{mle}-\theta_i^c)(\theta_j^{mle}-\theta_j^c)(\theta_p^{mle}-\theta_p^c)\right]| \to 0$ as $m \to \infty$ for $i,j,p = 1,2,\ldots,k$. Thus, $m\mathrm{E}\left[(\theta_i^{mle}-\theta_i^c)(\theta_j^{mle}-\theta_j^c)(\theta_p^{mle}-\theta_p^c)\right] \to 0$ as $m \to \infty$ for $i,j,p = 1,2,\ldots,k$. Therefore $m \times (A.5)$ converges to 0 and thus the scaled remainder $m\mathrm{E}\left[\Upsilon_3(\boldsymbol{\theta}^{mle})\right] \to 0$ as $m \to \infty$. All components of $mc(\boldsymbol{\theta}^{mle})$ converge to 0 as $m \to \infty$ as required.

□

A.3 Asymptotics of $\widehat{b}$

Our delta approximation of bias is $b^{approx} = \frac{1}{2}\,\mathrm{tr}(\Omega\mathrm{H}(\boldsymbol{\theta}^c))$, where $\mathrm{H}(\boldsymbol{\theta}^c)$ is the Hessian matrix of second-order partial derivatives of $\eta(\cdot)$ evaluated at $\boldsymbol{\theta}^c$ and $\Omega = \mathrm{Var}(\boldsymbol{\theta}^{mle})$, the variance-covariance matrix of the MLEs. Due to the unknowns in b^{approx} we estimate it by $\widehat{b} = \frac{1}{2}\mathrm{tr}(\widehat{\Omega}\widehat{\mathrm{H}}(\boldsymbol{\theta}^{mle}))$. We now show that, under certain assumptions, $m\widehat{b}$ converges to $mb^{approx} = \frac{1}{2}\mathrm{tr}(\mathrm{I}_0(\boldsymbol{\theta}^c)^{-1}\mathrm{H}(\boldsymbol{\theta}^c))$.

ASSUMPTION A.3: The expected simulation response, $\eta : \mathbb{R}^k \to \mathbb{R}$, is quadratic; i.e.,

$$\eta(\boldsymbol{\theta}) = \beta_0 + \boldsymbol{\theta}^T\boldsymbol{\beta} + \boldsymbol{\theta}^T\mathbf{B}\boldsymbol{\theta}. \tag{A.8}$$

ASSUMPTION A.4: Except for the point at which it is centered, the CCD is fixed and sufficient to support Model (A.8) such that $\widehat{\mathrm{B}}_{ij} \in \mathbb{R}$, the least squares estimator of B_{ij} is a consistent estimator for $i,j = 1,2,\ldots,k$. That is, $\widehat{\mathrm{B}}_{ij} \xrightarrow{P} \mathrm{B}_{ij}$ as $r \to \infty$ for $i,j = 1,2,\ldots,k$.

ASSUMPTION A.5: Let the simulation be driven by L independent parametric input distributions, with $k \geq L$ input parameters. Assume we have m observations from each of the

L distributions. Now let $\boldsymbol{\theta}^{mle} \in \mathbb{R}^k$ be the vector of MLEs given the m observations of each input distribution. We assume that

1. The MLEs are consistent, $\theta_i^{mle} \xrightarrow{P} \theta_i^c$ as $m \to \infty$ for $i = 1, 2, \ldots, k$.
2. The scaled variance of the MLEs $m\Omega$ tends to the inverse Fisher information at $\boldsymbol{\theta}^c$, $\mathrm{I}_0(\boldsymbol{\theta}^c)^{-1}$, as $m \to \infty$, $m\Omega \to \mathrm{I}_0(\boldsymbol{\theta}^c)^{-1}$ as $m \to \infty$.
3. The inverse Fisher information, $\mathrm{I}_0(\cdot)^{-1}$, is continuous.

THEOREM A.2: Let Assumptions A.3, A.4 and A.5 hold. Then the scaled estimate of the delta approximation of bias, $m\widehat{b}$, converges to the scaled delta approximation of bias; that is, as $m, r \to \infty$

$$m\widehat{b} \xrightarrow{P} \frac{1}{2}\mathrm{tr}(\mathrm{I}_0(\boldsymbol{\theta}^c)^{-1}\mathrm{H}(\boldsymbol{\theta}^c)).$$

Proof. First consider the Hessian. Under Assumption A.3 the expected simulation response is globally quadratic; therefore the Hessian does not depend on where we evaluate it since

$$\mathrm{H}(\boldsymbol{\theta}) = \begin{pmatrix} 2B_{11} & B_{12} & \ldots & B_{1k} \\ B_{21} & 2B_{22} & & \\ \vdots & & \ddots & \\ B_{k1} & & & 2B_{kk} \end{pmatrix}.$$

Thus $\widehat{b} = \frac{1}{2}\mathrm{tr}(\widehat{\Omega}\mathrm{H}(\boldsymbol{\theta}^{mle}))$ and this proof is equivalent to showing that $m\widehat{\Omega}\mathrm{H}(\boldsymbol{\theta}^{mle}) \xrightarrow{\mathrm{P}} \mathrm{I}_0(\boldsymbol{\theta}^c)^{-1}\mathrm{H}(\boldsymbol{\theta}^c)$.

Further, the least-squares estimators of the second-order terms are unchanged by shifting the center point of the design. Thus, under Assumption A.4, by completing r replications of the simulation at each of the design points of the CCD we gain the consistent estimators of the second-order partial derivatives, $\widehat{\mathrm{B}}_{ij} \xrightarrow{P} \mathrm{B}_{ij}$ for $i, j = 1, 2, \ldots, k$, such that $\mathrm{H}(\boldsymbol{\theta}) \xrightarrow{P} \mathrm{H}(\boldsymbol{\theta}^c)$ as $r \to \infty$ for any $\boldsymbol{\theta}$. Therefore, $\mathrm{H}(\boldsymbol{\theta}^{mle}) \xrightarrow{P} \mathrm{H}(\boldsymbol{\theta}^c)$ as $r \to \infty$.

Now consider $\widehat{\Omega} = \widehat{\mathrm{Var}}(\boldsymbol{\theta}^{mle})$. In practice we use the plug in estimator $\widehat{\Omega} = \mathrm{I}_0(\boldsymbol{\theta}^{mle})^{-1}/m$. Under Assumption A.5.1 and A.5.3, using continuous mapping theorem, $\mathrm{I}_0(\boldsymbol{\theta}^{mle})^{-1} \xrightarrow{\mathrm{P}} \mathrm{I}_0(\boldsymbol{\theta}^c)^{-1}$ as $m \to \infty$ thus $m\widehat{\Omega} \xrightarrow{\mathrm{P}} \mathrm{I}_0(\boldsymbol{\theta}^c)^{-1}$ as $m \to \infty$

Finally, by applying Slutsky's theorem we have $m\widehat{\Omega}\mathrm{H}(\boldsymbol{\theta}^{mle}) \xrightarrow{P} \mathrm{I}_0(\boldsymbol{\theta}^c)^{-1}\mathrm{H}(\boldsymbol{\theta}^c)$ as $m, r \to \infty$ as required.

□

REMARK 1: The results of Theorem A.1 and Theorem A.2 can be extended to the case where $m_1 \neq m_2 \neq \cdots \neq m_L$ provided that $m_i / \sum_{j=1}^{L} m_j \to c_i > 0$, for some fixed values c_i.

Appendix

B Results Tables - A Spline Function Method for Modelling and Generating a Nonhomogeneous Poisson Process

Table B.1: In $G = 500$ fits of the arrival rate function, the proportion of times the spline-based input model, "SPL", achieved the smallest maximum gap, ζ, or integrated absolute gap, δ, compared to the piecewise-linear, "PL", and piecewise-quadratic, "PQ", input models.

m_d	κ	ξ	ζ		δ	
			PQ/SPL	PL/SPL	PQ/SPL	PL/SPL
15	1	1	0.996	0.968	0.670	0.980
15	3	1	0.618	0.708	0.702	0.978
15	5	1	0.688	0.624	0.774	0.976
15	1	5	0.998	0.984	0.716	0.984
15	3	5	1.000	0.966	0.818	1.000
15	5	5	1.000	0.984	0.786	0.996
15	1	10	1.000	0.986	0.734	0.982
15	3	10	1.000	0.990	0.730	0.990
15	5	10	1.000	0.980	0.798	0.990
30	1	1	1.000	0.934	0.794	0.988
30	3	1	0.692	0.660	0.800	0.996
30	5	1	0.760	0.662	0.860	0.992
30	1	5	1.000	0.996	0.762	0.998
30	3	5	0.998	0.978	0.772	0.994
30	5	5	1.000	0.992	0.784	0.998
30	1	10	1.000	0.982	0.772	0.978
30	3	10	1.000	0.994	0.794	0.998
30	5	10	0.998	0.994	0.824	0.992
100	1	1	0.992	0.896	0.868	0.998
100	3	1	0.832	0.732	0.850	0.998
100	5	1	0.992	0.990	0.808	1.000
100	1	5	1.000	0.994	0.802	0.994
100	3	5	0.996	1.000	0.832	0.998
100	5	5	0.988	0.996	0.852	0.996
100	1	10	1.000	0.998	0.822	0.998
100	3	10	1.000	0.996	0.856	1.000
100	5	10	0.998	0.994	0.888	1.000

Table B.2: Given underdispersed observations with target $cv = 0.5^2$. In $G = 500$ fits of the arrival rate function, the proportion of times the spline-based input model, "SPL", achieved the smallest maximum gap, ζ, or integrated absolute gap, δ, compared to the piecewise-linear, "PL", and piecewise-quadratic, "PQ", input models.

m_d	κ	ξ	ζ		δ	
			PQ/SPL	PL/SPL	PQ/SPL	PL/SPL
15	1	1	0.996	0.962	0.620	0.972
15	3	1	0.566	0.746	0.642	0.990
15	5	1	0.870	0.882	0.578	0.972
15	1	5	1.000	1.000	0.768	0.998
15	3	5	1.000	1.000	0.922	1.000
15	5	5	1.000	0.998	0.718	0.998
15	1	10	1.000	0.986	0.746	0.994
15	3	10	1.000	0.998	0.828	0.994
15	5	10	1.000	0.998	0.796	0.996
30	1	1	1.000	0.918	0.796	0.996
30	3	1	0.702	0.780	0.760	0.998
30	5	1	0.978	0.934	0.710	0.992
30	1	5	1.000	0.996	0.800	0.994
30	3	5	1.000	0.998	0.798	0.998
30	5	5	1.000	0.994	0.698	1.000
30	1	10	1.000	0.986	0.790	0.992
30	3	10	1.000	0.996	0.818	1.000
30	5	10	1.000	0.992	0.808	0.994
100	1	1	1.000	0.862	0.86	0.988
100	3	1	0.996	0.966	0.766	0.998
100	5	1	1.000	1.000	0.938	1.000
100	1	5	1.000	0.994	0.824	0.994
100	3	5	0.996	0.988	0.708	0.998
100	5	5	0.956	0.986	0.756	0.992
100	1	10	1.000	0.984	0.824	0.994
100	3	10	1.000	0.986	0.808	1.000
100	5	10	0.990	0.988	0.878	0.998

Table B.3: Given overdispersed observations with target $cv = 1.5^2$. In $G = 500$ fits of the arrival rate function, the proportion of times the spline-based input model, “SPL”, achieved the smallest maximum gap, ζ, or integrated absolute gap, δ, compared to the piecewise-linear, “PL”, and piecewise-quadratic, “PQ”, input models.

m_d	κ	ξ	ζ		δ	
			PQ/SPL	PL/SPL	PQ/SPL	PL/SPL
15	1	1	1.000	0.996	0.836	0.990
15	3	1	0.662	0.826	0.854	0.964
15	5	1	0.662	0.762	0.930	0.986
15	1	5	1.000	0.988	0.878	0.990
15	3	5	0.994	0.978	0.924	0.992
15	5	5	1.000	0.978	0.926	0.996
15	1	10	1.000	0.994	0.888	0.988
15	3	10	1.000	0.998	0.886	0.990
15	5	10	1.000	0.998	0.934	1.000
30	1	1	0.998	0.974	0.902	0.988
30	3	1	0.662	0.792	0.922	0.988
30	5	1	0.722	0.784	0.964	1.000
30	1	5	1.000	0.986	0.892	0.986
30	3	5	1.000	0.984	0.886	0.990
30	5	5	0.998	0.992	0.944	1.000
30	1	10	1.000	0.992	0.868	0.990
30	3	10	1.000	0.990	0.930	0.996
30	5	10	1.000	0.998	0.952	0.998
100	1	1	0.994	0.960	0.950	0.996
100	3	1	0.698	0.854	0.966	0.998
100	5	1	0.984	0.990	0.960	1.000
100	1	5	0.998	0.994	0.908	0.998
100	3	5	0.996	0.998	0.942	1.000
100	5	5	0.992	0.998	0.954	1.000
100	1	10	1.000	0.998	0.93	0.996
100	3	10	1.000	1.000	0.956	1.000
100	5	10	1.000	1.000	0.980	0.998

Bibliography

Abramowitz, M. and Stegun, I. A. (1964). *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*, volume 55. Courier Corporation.

Ankenman, B. E. and Nelson, B. L. (2012). A Quick Assessment of Input Uncertainty. In Laroque, C., Himmelspach, J., Pasupathy, R., Rose, O., and Uhrmacher, A. M., editors, *Proceedings of the 2012 Winter Simulation Conference*, pages 1–10, Piscataway, New Jersey. IEEE Press.

Arena (2015). Arena. Rockwell Automation. https://www.arenasimulation.com/.

Arkin, B. L. and Leemis, L. M. (2000). Nonparametric Estimation of the Cumulative Intensity Function for a Nonhomogeneous Poisson Process from Overlapping Realizations. *Management Science*, 46(7):989–998.

Avramidis, A. N., Deslauriers, A., and L'Ecuyer, P. (2004). Modeling Daily Arrivals to a Telephone Call Center. *Management Science*, 50(7):896–908.

Banks, J., Carson, J. S., Nelson, B. L., and Nicol, D. M. (2013). *Discrete-Event System Simulation: Pearson New International Edition*. Pearson Higher Ed.

Barton, R. R. (2012). Tutorial: Input Uncertainty in Outout Analysis. In Laroque, C., Himmelspach, J., Pasupathy, R., Rose, O., and Uhrmacher, A., editors, *Proceedings of the 2012 Winter Simulation Conference*, pages 1–12, Piscataway, New Jersey. IEEE Press.

Barton, R. R., Nelson, B. L., and Xie, W. (2010). A Framework for Input Uncertainty

Analysis. In *Proceedings of the 2010 Winter Simulation Conference*, pages 1189–1198. Winter Simulation Conference.

Barton, R. R., Nelson, B. L., and Xie, W. (2013). Quantifying Input Uncertainty via Simulation Confidence Intervals. *INFORMS Journal on Computing*, 26(1):74–87.

Barton, R. R. and Schruben, L. W. (1993). Uniform and Bootstrap Resampling of Empirical Distributions. In *Proceedings of the 1993 Winter Simulation Conference*, pages 503–508. ACM.

Barton, R. R. and Schruben, L. W. (2001). Resampling Methods for Input Modeling. In *Proceedings of the 2001 Winter Simulation Conference*, pages 372–378. IEEE Computer Society.

Biller, B. and Corlu, C. G. (2011). Accounting for Parameter Uncertainty in Large-Scale Stochastic Simulations with Correlated Inputs. *Operations Research*, 59(3):661–673.

Blumenfeld, D. (2009). *Operations Research Calculations Handbook*. CRC Press.

Box, G. E. (1978). Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building. *Probability and Mathematical Statistics*.

Brailsford, S. C. (2007). Advances and Challenges in Healthcare Simulation Modeling: Tutorial. In *Proceedings of the 2007 Winter Simulation Conference*, pages 1436–1448. IEEE Press.

Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Shen, H., Zeltyn, S., and Zhao, L. (2005). Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective. *Journal of the American statistical association*, 100(469):36–50.

Cavanaugh, J. E. and Neath, A. A. (2011). Akaike's Information Criterion: Background, Derivation, Properties, and Refinements. In *International Encyclopedia of Statistical Science*, pages 26–29. Springer.

Channouf, N. (2008). *Modélisation et Optimisation d'un Centre d'appels Téléphoniques: étude du Processus d'arrivée*. PhD thesis, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Montréal.

Chen, H. and Schmeiser, B. (2013). I-SMOOTH: Iteratively Smoothing Mean-Constrained and Nonnegative Piecewise-Constant Functions. *INFORMS JoC*, 25(3):432–445.

Chen, H. and Schmeiser, B. W. (2017). MNO–PQRS: Max Nonnegativity Ordering — Piecewise-Quadratic Rate Smoothing. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 27(3):18.

Chen, H. and Schmeiser, B. W. (2018). MISE-Optimal Grouping of Point-Process Data with a Constant Dispersion Ratio. In *Proceedings of the 2018 Winter Simulation Conference*. IEEE Press.

Chen, J. and Gupta, A. K. (2011). *Parametric Statistical Change Point Analysis: With Applications to Genetics, Medicine, and Finance*. Springer Science & Business Media.

Cheng, R. (2017). History of Input Modeling. In *Proceedings of the 2017 Winter Simulation Conference*, pages 181–201. IEEE Press.

Cheng, R. C. (1994). Selecting Input Models. In *Proceedings of the 1994 Winter Simulation Conference*, pages 184–191, Piscataway, New Jersey. Society for Computer Simulation International.

Cheng, R. C. and Holland, W. (1997). Sensitivity of Computer Simulation Experiments to Errors in Input Data. *Journal of Statistical Computation and Simulation*, 57(1-4):219–241.

Cheng, R. C. and Holland, W. (1998). Two-point Methods for Assessing Variability in Simulation Output. *Journal of Statistical Computation Simulation*, 60(3):183–205.

Cheng, R. C. and Holland, W. (2004). Calculation of Confidence Intervals for Simulation Output. *ACM Transactions on Modeling and Computer Simulation*, 14:344–362.

Chernick, M. R. (2008). *Bootstrap methods: A guide for practitioners and researchers*. Nj: Wiley.

Chick, S. E. (1997). Bayesian Analysis for Simulation Input and Output. In *Proceedings of the 1997 Winter Simulation Conference*, pages 253–260. IEEE Computer Society.

Chick, S. E. (2001). Input Distribution Selection for Simulation Experiments: Accounting for Input Uncertainty. *Operations Research*, 49(5):744–758.

Cinlar, E. (2013). *Introduction to Stochastic Processes*. Courier Corporation.

Conn, A. R., Gould, N. I., and Toint, P. L. (2000). *Trust Region Methods*, volume 1. Siam.

Cox, D. and A. W. Lewis, P. (1966). *The Statistical Analysis of Series of Events*. Chapman and Hall.

de Boor, C. (1978). *A Practical Guide to Splines*, volume 27. Springer-Verlag New York.

Dean, A. and Voss, D. (1999). *Response Surface Methodology*. Springer-Verlag, New York.

Dixon, M. F. and Ward, T. (2018). Takeuchi's Information Criteria as a Form of Regularization. https://arxiv.org/pdf/1803.04947.pdf.

Efron, B. (1982). *The Jackknife, the Bootstrap, and other Resampling Plans*, volume 38. Siam.

Efron, B. and Tibshirani, R. (1986). Bootstrap Methods for Standard Errors, Confidence Intervals, and other Measures of Statistical Accuracy. *Statistical Science*, pages 54–75.

Efron, B. and Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. CRC press.

Eliers, P. and Marx, B. (1996). Flexible Smoothing using B-Splines and Penalized Likelihood. *Statistical Science*, 11:1200–1224.

Gerhardt, I. and Nelson, B. L. (2009). Transforming Renewal Processes for Simulation of Nonstationary Arrival Processes. *INFORMS Journal on Computing*, 21(4):630–640.

Gray, R. J. (1992). Flexible Methods for Analyzing Survival Data using Splines, with Applications to Breast Cancer Prognosis. *Journal of the American Statistical Association*, 87(420):942–951.

Green, L. (2006). Queueing Analysis in Healthcare. In *Patient Flow: Reducing Delay in Healthcare Delivery*, pages 281–307. Springer.

Gurobi Optimization, L. (2018). Gurobi Optimizer Reference Manual.

Hardy, G. H., Littlewood, J. E., and Pólya, G. (1952). *Inequalities*. Cambridge University Press.

Henderson, S. G. (2003). Estimation for Nonhomogeneous Poisson Processes from Aggregated Data. *Operations Research Letters*, 31(5):375–382.

Kao, E. P. and Chang, S.-L. (1988). Modeling Time-Dependent Arrivals to Service Systems: A Case in using a Piecewise-Polynomial Rate Function in a Nonhomogeneous Poisson Process. *Management Science*, 34(11):1367–1379.

Kim, S.-H. and Whitt, W. (2014). Are Call Center and Hospital Arrivals Well Modeled by Nonhomogeneous Poisson Processes? *Manufacturing & Service Operations Management*, 16(3):464–480.

Kingman, J. F. C. (1992). *Poisson Processes*, volume 3. Clarendon Press.

Klein, R. W. and Roberts, S. D. (1984). A Time-Varying Poisson Arrival Process Generator. *Simulation*, 43(4):193–195.

Kuhl, M. E. and Bhairgond, P. S. (2000). Nonparametric Estimation of Nonhomogeneous Poisson Processes using Wavelets. In Joines, J., Barton, R., Kang, K., and Fishwick, P., editors, *Proceedings of the 2000 Winter Simulation Conference*, Piscataway, New Jersey. IEEE Press.

Kuhl, M. E. and Wilson, J. R. (2000). Least Squares Estimation of Nonhomogeneous Poisson Processes. *Journal of Statistical Computation and Simulation*, 67(1):699–712.

Kuhl, M. E. and Wilson, J. R. (2009). Advances in Modeling and Simulation of Nonstationary Arrival Processes. In *Proceedings of the 2009 INFORMS SSR Workshop*, pages 1–5.

Kuhl, M. E., Wilson, J. R., and Johnson, M. A. (1995). Estimation and Simulation of Nonhomogeneous Poisson Processes having Multiple Periodicities. In *Proceedings of the 1995 Winter Simulation Conference*, pages 374–383. IEEE Computer Society.

Kuhl, M. E., Wilson, J. R., and Johnson, M. A. (1997). Estimating and Simulating Poisson Processes having Trends or Multiple Periodicities. *IIE transactions*, 29(3):201–211.

Kullback, S. (1997). *Information Theory and Statistics*. Courier Corporation.

Law, A. M. (1988). Simulation of Manufacturing Systems. In *Proceedings of the 1988 Winter Simulation Conference*, pages 40–51. IEEE Press.

Lee, S., Wilson, J. R., and Crawford, M. M. (1991). Modeling and Simulation of a Nonhomogeneous Poisson Process having Cyclic Behavior. *Communications in Statistics-Simulation and Computation*, 20(2-3):777–809.

Leemis, L. (2001). Input Modeling: Input Modeling Techniques for Discrete-Event Simulations. In *Proceedings of the 2001 Winter Simulation Conference*, pages 62–73, Piscataway, New Jersey. IEEE Computer Society.

Leemis, L. M. (1991). Nonparametric Estimation of the Cumulative Intensity Function for a Nonhomogeneous Poisson Process. *Management Science*, 37(7):886–900.

Leemis, L. M. (2004). Nonparametric Estimation and Variate Generation for a Nonhomogeneous Poisson Process from Event Count Data. *IIE Transactions*, 36(12):1155–1160.

Lewis, P. A. (1971). Recent Results in the Statistical Analysis of Univariate Point Processes. Technical report, Monterey, California. Naval Postgraduate School.

Lewis, P. A. and Shedler, G. S. (1976). Statistical Analysis of Non-stationary Series of Events in a Data Base System. Technical report, Naval Postgraduate School Montery CA.

Lin, Y., Song, E., and Nelson, B. L. (2015). Single-Experiment Input Uncertainty. *Journal of Simulation*, 9:249—-259.

Massey, W. A., Parker, G. A., and Whitt, W. (1996). Estimating the Parameters of a Non-homogeneous Poisson Process with Linear Rate. *Telecommunication Systems*, 5(2):361–388.

Montgomery, D. C. (2013). *Design and Analysis of Experiments*. John Wiley & Sons.

Morgan, L. E., Titman, A. C., Worthington, D. J., and Nelson, B. L. (2016). Input Uncertainty Quantification for Simulation Models with Piecewise-constant Non-stationary Poisson Arrival Processes. In *Proceedings of the 2016 Winter Simulation Conference*, pages 370–381. IEEE Press.

Morgan, L. E., Titman, A. C., Worthington, D. J., and Nelson, B. L. (2017). Detecting Bias due to Input Modelling in Computer Simulation. In *Proceedings of the 2017 Winter Simulation Conference*, pages 1974–1985. IEEE Press.

Nelson, B. (2013). *Foundations and Methods of Stochastic Simulation: A First Course*. Springer Science & Business Media.

Nelson, B. L. and Gerhardt, I. (2011). Modelling and Simulating Non-stationary Arrival Processes to Facilitate Analysis. *Journal of Simulation*, 5(1):3–8.

Nelson, B. L. and Yamnitsky, M. (1998). Input Modeling Tools for Complex Problems. In *Proceedings of the 1998 Winter Simulation Conference*, pages 105–112. IEEE Computer Society Press.

Ng, S. H. and Chick, S. E. (2001). Reducing Input Parameter Uncertainty for Simulations. In *Proceedings of the 2001 Winter Simulation Conference*, pages 364–371. IEEE Computer Society.

Ng, S. H. and Chick, S. E. (2006). Reducing Parameter Uncertainty for Stochastic Systems. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 16(1):26–51.

Nicol, D. M. and Leemis, L. M. (2014). A Continuous Piecewise-Linear NHPP Intensity Function Estimator. In *Proceedings of the 2014 Winter Simulation Conference*, pages 498–509, Piscataway, New Jersey. IEEE Press.

Oehlert, G. W. (1992). A Note on the Delta Method. *The American Statistician*, 46(1):27–29.

Oreshkin, B. N., Réegnard, N., and L'Ecuyer, P. (2016). Rate-Based Daily Arrival Process Models with Application to Call Centers. *Operations Research*, 64(2):510–527.

Osius, G. (1989). Some Results on Convergence of Moments and Convergence in Distribution with Applications in Statistics. *Mathematik-Arbeitspapiere No. 33*.

Pritsker, A., Martin, D., Reust, J., Wagner, M., Wilson, J., Kuhl, M., Roberts, J., Daily, O., Harper, A., Edwards, E., et al. (1996). Organ Transplantation Modeling and Analysis. In *Proceedings of the 1996 Western Multiconference: Simulation in the Medical Sciences*, pages 29–35. The Society for Computer Simulation, San Diego, CA.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Reinsch, C. H. (1967). Smoothing by Spline Functions. *Numerische mathematik*, 10(3):177–183.

Rhodes-Leader, L., Worthington, D. J., Nelson, B. L., and Onggo, B. S. (2018). Multi-Fidelity Simulation Optimisation for Airline Disruption Management. In *Proceedings of the 2018 Winter Simulation Conference*. IEEE Press.

Sanchez, S. M. and Sanchez, P. J. (2005). Very Large Fractional Factorial and Central Composite Designs. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 15(4):362–377.

Schoenberg, I. J. (1964). Spline Functions and the Problem of Graduation. *Proceedings of the National Academy of Sciences*, 52(4):947–950.

Sellers, K. F. and Morris, D. S. (2017). Underdispersion Models: Models that are "Under the Radar". *Communications in Statistics-Theory and Methods*, 46(24):12075–12086.

Shibata, R. (1989). Statistical Aspects of Model Selection. In *From Data to Model*, pages 215–240. Springer.

Shikin, E. V. and Plis, A. I. (1995). *Handbook on Splines for the User*. CRC Press.

Simio (2015). SIMIO. Simio LLC. www.simio.com/.

SIMUL8 (2015). SIMUL8. Simul8 Corporation. http://www.simul8.com/.

Song, E. and Nelson, B. L. (2013). A Quicker Assessment of Input Uncertainty. In *Proceedings of the 2013 Winter Simulation Conference*, pages 474–485. IEEE Press.

Song, E. and Nelson, B. L. (2015). Quickly Assessing Contributions to Input Uncertainty. *IIE Transactions*, 47:893–909.

Song, E., Nelson, B. L., and Pegden, C. D. (2014). Advanced Tutorial: Input Uncertainty Quantification. In Tolk, A., Diallo, S. Y., Ryzhov, I. O., Yilmaz, L., Buckley, S., and Miller, J. A., editors, *Proceedings of the 2014 Winter Simulation Conference*, pages 162–176, Piscataway, New Jersey. IEEE Press.

Viswanadham, N. and Narahari, Y. (1992). *Performance Modeling of Automated Manufacturing Systems*. Prentice Hall Englewood Cliffs, NJ.

Whitt, W. (2007). What you should know about Queueing Models to set Staffing Requirements in Service Systems. *Naval Research Logistics (NRL)*, 54(5):476–484.

Whittaker, E. T. (1922). On a new Method of Graduation. *Proceedings of the Edinburgh Mathematical Society*, 41:63–75.

Wieland, J. R. and Schmeiser, B. W. (2006). Stochastic Gradient Estimation Using a Single Design Point. In *Proceedings of the 2006 Winter Simulation Conference*, pages 390–397, Piscataway, New Jersey. IEEE Press.

Winkelbauer, A. (2012). Moments and Absolute Moments of the Normal Distribution. *arXiv preprint arXiv:1209.4340*.

Withers, C. S. and Nadarajah, S. (2014). Bias Reduction: The Delta Method versus the Jackknife and the Bootstrap. *Pakistan Journal of Statistics*, 30(1):143–151.

Wright, S. and Nocedal, J. (1999). *Numerical Optimization*, volume 35. Springer.

Xie, W., Nelson, B. L., and Barton, R. R. (2014a). A Bayesian Framework for Quantifying Uncertainty in Stochastic Simulation. *Operations Research*, 62(6):1439–1452.

Xie, W., Nelson, B. L., and Barton, R. R. (2014b). Statistical Uncertainty Analysis for Stochastic Simulation with Dependent Input Models. In *Proceedings of the 2014 Winter Simulation Conference*, pages 674–685. IEEE Press.

Xie, W., Nelson, B. L., and Barton, R. R. (2016). Multivariate Input Uncertainty in Output Analysis for Stochastic Simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 27(1):5.

Zheng, Z. and Glynn, P. W. (2017). Fitting Continuous Piecewise Linear Poisson Intensities via Maximum Likelihood and Least Squares. In *Proceedings of the 2017 Winter Simulation Conference*, pages 1740–1749. IEEE Press.

Zouaoui, F. and Wilson, J. R. (2003). Accounting for Parameter Uncertainty in Simulation Input Modeling. *IIE Transactions*, 35(9):781–792.

Zouaoui, F. and Wilson, J. R. (2004). Accounting for Input-model and Input-parameter Uncertainties in Simulation. *IIE Transactions*, 36(11):1135–1151.

Zouaoui, F. and Wilson, J. R. (2010). Accounting for Input-model and Input-parameter uncertainties in Simulation. *IIE Transactions*, 36(11):1135–1151.