Methodol Comput Appl Probab (2008) 10:239–275
DOI 10.1007/s11009-007-9045-3

Quasi-Monte Carlo for Highly Structured Generalised Response Models

F. Y. Kuo · W. T. M. Dunsmuir · I. H. Sloan · M. P. Wand · R. S. Womersley

Received: 6 November 2006 / Revised: 11 July 2007 /
Accepted: 12 July 2007 / Published online: 4 October 2007

Abstract Highly structured generalised response models, such as generalised linear mixed models and generalised linear models for time series regression, have become an indispensable vehicle for data analysis and inference in many areas of application. However, their use in practice is hindered by high-dimensional intractable integrals. Quasi-Monte Carlo (QMC) is a dynamic research area in the general problem of high-dimensional numerical integration, although its potential for statistical applications is yet to be fully explored. We survey recent research in QMC, particularly lattice rules, and report on its application to highly structured generalised response models. New challenges for QMC are identified and new methodologies are developed. QMC methods are seen to provide significant improvements compared with ordinary Monte Carlo methods.

Keywords Generalised linear mixed models · High-dimensional integration · Lattice rules · Longitudinal data analysis · Maximum likelihood · Quasi-Monte Carlo · Semiparametric regression · Serial dependence · Time series regression

AMS 2000 Subject Classification 62J12 · 62M10 · 62M30 · 65D30 · 65D32

F. Y. Kuo · W. T. M. Dunsmuir (✉) · I. H. Sloan · M. P. Wand · R. S. Womersley
School of Mathematics and Statistics, University of New South Wales,
Sydney NSW 2052, Australia
e-mail: W.Dunsmuir@unsw.edu.au

F. Y. Kuo
e-mail: F.Kuo@unsw.edu.au

I. H. Sloan
e-mail: I.Sloan@unsw.edu.au

M. P. Wand
e-mail: Wand@maths.unsw.edu.au

R. S. Womersley
e-mail: R.Womersley@unsw.edu.au

1 Introduction

Messy data sets with a generalised response variable abound in many contemporary areas of application. By "generalised response" we mean a response variable that is far from normally distributed, such as a binary, count or heavily skewed variable. This departure from normality is the motivation for the development of generalised linear models and extensions. An example of such data is given in Fig. 1 and involves longitudinal measurements on 275 Indonesian children from Diggle et al. (1995). The response variable is an indicator of respiratory infection. Of interest are the effects of covariates such as vitamin A nourishment. However, regression-type analyses need to account for correlation among repeated measures on the same child as well as a possibly non-linear age effect.

A useful vehicle for analysis of data such as these is the set of generalised linear mixed models. A random intercept can take care of the within subject correlation. Provided the design matrices are permitted to be of general form then penalised spline basis functions can take care of non-linear covariate effects (e.g. Zhao et al. 2006). However, the likelihood involves intractable integrals of dimension as high as the number of basis functions, usually in the range 10–35. Model fitting and inference require strategies for dealing with such integrals. The most common approaches

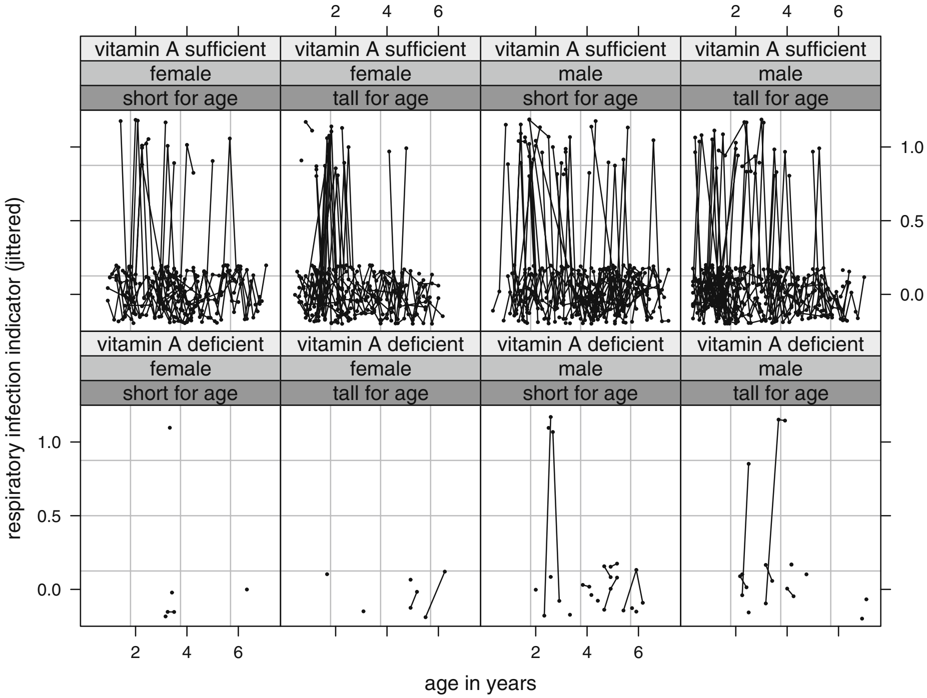

Fig. 1 Respiratory infection indicator (0 = absent, 1 = present) versus age in years for a cohort of Indonesian children. Repeated measures on same child are *connected by lines*. *Each panel* corresponds to a different combination of three covariates: vitamin A nourishment, gender and height for age. The respiratory infection indicator has been jittered to enhance visualisation

involve Laplace approximation (e.g. Breslow and Clayton 1993) and Markov Chain Monte Carlo or MCMC (e.g. Clayton 1996; Zhao et al. 2006). Kuk (1999) reviews Laplace importance sampling for generalised linear mixed models and relates this to use of MCMC procedures.

In time series regression with generalised response the integration problems are even worse. Davis et al. (2000) discuss several applications of Poisson regression in time series with serial dependence arising in public health: a series of $T = 168$ monthly polio counts and a series of $T = 1465$ daily asthma counts. The dimension of the integral, $d = T$, corresponds to the length of the time series—potentially in the hundreds or thousands. This is of the same order as a 360-dimensional integration problem in mathematical finance that was successfully handled by Paskov and Traub (1995). This phenomenal breakthrough was due to integration technology that has become known as *quasi-Monte Carlo* or *QMC*.

Like the Monte Carlo method, QMC methods approximate high-dimensional integrals by averages of function values sampled at a number of points. However, instead of generating the sample points randomly, the QMC integration points are chosen deterministically in a clever way to be “better than random”: they are designed to achieve a faster rate of convergence, thus are more effective and more efficient in practice.

In this article we investigate the extent to which QMC can handle high-dimensional integrals arising in highly structured generalised response models. A necessary step before applying any QMC algorithm is to transform the integrand into the unit cube. It is seen that this transformation, which relates closely to the technique of importance sampling, plays a crucial role in QMC integration. Extensive numerical studies show that QMC does have something to offer, in comparison to ordinary Monte Carlo integration, for highly structured generalised response models.

Hickernell et al. (2005) contains a recent survey of QMC methodology for a statistical readership. Recent applications of QMC to generalised linear mixed models have mainly focused on the binomial or binary responses via a logistic regression model with various correlation or random effects structures. Pan and Thompson (1998, 2004, 2007) and Al-Eid and Pan (2005) describe the use of QMC for generalised linear mixed model fitting. They compare the application of QMC in maximum likelihood with Gibb’s sampling for modelling the well-known Salamander Mating data in which $d = 20$, and report that QMC is computationally much faster. González et al. (2006) apply QMC methods for maximum likelihood estimation with $d \leq 9$ clustered random effects and report improved accuracy and computing time using QMC. Jank (2005) has investigated the use of randomized QMC methods in implementing the Monte Carlo expectation-maximisation (EM) algorithm and demonstrates, using a spatial correlation structure with integral dimension $d = 16$ in a geostatistical model of on-line purchases, that the use of QMC can significantly reduce the simulation effort compared with classical Monte Carlo EM. In contrast to these recent applications to binary response models in this paper we focus on Poisson response models in which $d \geq 25$.

Section 2 provides a summary of highly structured generalised response models and the integration challenges that the log-likelihood presents. Section 3 surveys QMC from its elements up to its most recent developments. QMC approaches to the aforementioned log-likelihood integrals are described in Section 4. Numerical experiments are documented in Section 5. A discussion of the optimisation procedure is given in Section 6. Section 7 contains a brief summary and the Appendix contains additional technical details.

2 Highly Structured Generalised Response Models

Let $\mathbf{y}$ be a generalised response vector for which a model in terms of several predictors is sought. The most common generalised response situations are binary $\mathbf{y}$, which results in a conditional Bernoulli likelihood structure; and $\mathbf{y}$ containing counts, in which case conditional Poisson likelihood structure is often assumed. A rich class of such models is

$$\begin{aligned} f(\mathbf{y}|\mathbf{w}) &= \exp\left\{\mathbf{y}^\top(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w}) - \mathbf{1}^\top b\,(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w}) + \mathbf{1}^\top c(\mathbf{y})\right\}, \\ \mathbf{w} &\sim N(\mathbf{0},\boldsymbol{\Sigma}), \end{aligned} \tag{1}$$

where here, and throughout, we use the convention $s(\mathbf{v}) \equiv (s(v_1),\ldots,s(v_n))^\top$ for a scalar-valued function s and an $n\times 1$ vector $\mathbf{v} = (v_1,\ldots,v_n)^\top$. Setting $b(x) = \log(1+e^x)$ and $c(x) = 0$ in (1) gives the Bernoulli conditional density with conditional mean $\mathsf{E}(\mathbf{y}|\mathbf{w}) = \exp(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w})/\{\mathbf{1}+\exp(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w})\}$. The Poisson conditional density with conditional mean $\mathsf{E}(\mathbf{y}|\mathbf{w}) = \exp(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w})$ corresponds to $b(x) = e^x$ and $c(x) = \log(1/x!)$. The model parameters are $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$ although it is common to impose further parametric structure on $\boldsymbol{\Sigma}$. The matrix $\mathbf{X}$ is a design matrix corresponding to the effects in $\boldsymbol{\beta}$. The matrix $\mathbf{W}$ could be a design matrix or simply the identity matrix of appropriate dimension. Since there is a wide range of choices for $\mathbf{W}$ and $\boldsymbol{\Sigma}$ we call such models *highly structured*. Special cases include generalised linear mixed models (e.g. McCulloch and Searle 2000) and generalised linear models for time series regression (e.g. Davis et al. 1999). Sections 2.1–2.3 provide specific illustrations.

Models of the form (1) are typically fitted by using maximum likelihood or hierarchical Bayes methodology. We will focus on the likelihood approach, although the ideas are extendible to Bayesian approaches, as discussed below. The log-likelihood of $(\boldsymbol{\beta},\boldsymbol{\Sigma})$ is

$$\begin{aligned} \ell(\boldsymbol{\beta},\boldsymbol{\Sigma}) = \log\int_{\mathbb{R}^d} \exp\left\{\mathbf{y}^\top(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w}) - \mathbf{1}^\top b\,(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w}) - \frac{1}{2}\mathbf{w}^\top\boldsymbol{\Sigma}^{-1}\mathbf{w}\right\} d\mathbf{w} \\ -\frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{d}{2}\log(2\pi) + \mathbf{1}^\top c(\mathbf{y}), \end{aligned} \tag{2}$$

where d is the dimension of $\mathbf{w}$. However, its computation is thwarted by the intractable integral comprising the first term. By the end of this section we will have described some situations where the integral factorises into integrals of dimension of about 1–3; others that require integration over dimensions in the tens; and some that require integration over a space having the same dimension as the sample size. The second and third situations pose an obvious challenge.

A simple, and often useful, approximation to $\ell(\boldsymbol{\beta},\boldsymbol{\Sigma})$ arises from the Laplace method. This involves a multivariate normal approximation to the integrand via a second order Taylor series approximation of the exponent about its stationary point (e.g. Breslow and Clayton 1993). The resulting approximation is

$$\ell(\boldsymbol{\beta},\boldsymbol{\Sigma}) \approx \mathbf{y}^\top(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w}^*) - \mathbf{1}^\top b(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w}^*) - \frac{1}{2}\mathbf{w}^{*\top}\boldsymbol{\Sigma}^{-1}\mathbf{w}^*$$
$$-\frac{1}{2}\log|\mathbf{I}+\boldsymbol{\Sigma}\mathbf{W}^\top\mathrm{diag}\{b''(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w}^*)\}\mathbf{W}| + \mathbf{1}^\top c(\mathbf{y}),$$

where $\mathbf{w}^*$ is a solution to $\mathbf{w} = \boldsymbol{\Sigma}\mathbf{W}^\top\{\mathbf{y} - b'(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w})\}$. A commonly used further approximation for estimation of $\boldsymbol{\beta}$ is one which assumes that the determinant term is effectively constant as a function of $\boldsymbol{\beta}$ (Breslow and Clayton 1993).

The hierarchical Bayesian counterpart of (1) is

$$f(\mathbf{y}|\mathbf{w},\boldsymbol{\Sigma}) = \exp\left\{\mathbf{y}^\top(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w}) - \mathbf{1}^\top(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w}) + \mathbf{1}^\top c(\mathbf{y})\right\},$$
$$\mathbf{w}|\boldsymbol{\Sigma} \sim N(\mathbf{0},\boldsymbol{\Sigma}), \tag{3}$$

with appropriate prior distributions placed on $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$. Zhao et al. (2006) is a recent synopsis on the generalised linear mixed model (see Section 2.1). We are not aware of work on (3) in its full generality. The integrals that arise in Bayesian inference for (3) are on par with those required for frequentist inference for (1). Therefore, we anticipate that QMC methodology for the latter will be transferable to the former.

An alternative to maximising the likelihood directly is the use of the EM algorithm in which the E-step requires evaluation of similar high dimensional integrals. Caffo et al. (2005) review importance sampling and MCMC in Monte Carlo EM. Booth et al. (2001) discuss the relative merits of maximum likelihood, Monte Carlo EM, Monte Carlo Newton Raphson, and stochastic approximation. Owen and Tribble (2005) introduce a QMC version of the Metropolis-Hastings algorithm.

2.1 Generalised Linear Mixed Models

Generalised linear mixed models (GLMM) have become very popular in applied statistics due to their ability to handle a variety of complications arising in contemporary data analysis and inference. A survey of 20th Century work in the area is provided by McCulloch and Searle (2000). Recent books demonstrating the breadth of GLMM include Ruppert et al. (2003) and Skrondal and Rabe-Hesketh (2004).

An important class of GLMM corresponds to (1) with $\mathbf{w}$ set to $\mathbf{u}$, a random effects vector. The $\mathbf{W}$ matrix would then correspond to the design matrix attached to $\mathbf{u}$, usually denoted by $\mathbf{Z}$. Lastly, $\boldsymbol{\Sigma}$ corresponds to the covariance matrix of the random effects and is often denoted by $\mathbf{G}$. In this section we will use the $\mathbf{Z}$ and $\mathbf{G}$, rather than $\mathbf{W}$ and $\boldsymbol{\Sigma}$, to conform with the GLMM literature. The GLMM version of (1) is then

$$f(\mathbf{y}|\mathbf{u}) = \exp\left\{\mathbf{y}^\top(\mathbf{X}\boldsymbol{\beta}+\mathbf{Z}\mathbf{u}) - \mathbf{1}^\top b(\mathbf{X}\boldsymbol{\beta}+\mathbf{Z}\mathbf{u}) + \mathbf{1}^\top c(\mathbf{y})\right\},$$
$$\mathbf{u} \sim N(\mathbf{0},\mathbf{G}). \tag{4}$$

Distinguishing features between this model and its generalisation (1) are: the length of the random effects vector $\mathbf{u}$ is typically much smaller than that of the response vector $\mathbf{y}$, and the covariance matrix $\mathbf{G}$ has relatively simple structure and does contain forms arising in time series modelling, such as Toeplitz structure. For many important GLMMs, $\mathbf{G}$ is simply a diagonal matrix with only a few distinct diagonal entries.

Recently Zhao et al. (2006) examined Markov Chain Monte Carlo fitting of a hierarchical Bayes formulation of (4). They made a case for breaking up the linear predictor into sub-components that handle the various covariance structures used in longitudinal data modelling, smoothing and spatial statistics. Considered were decompositions of the design structure such as

$$\mathbf{X}\boldsymbol{\beta}+\mathbf{Z}\mathbf{u} = \mathbf{X}^{\mathrm{R}}\boldsymbol{\beta}^{\mathrm{R}}+\mathbf{Z}^{\mathrm{R}}\mathbf{u}^{\mathrm{R}}+\mathbf{X}^{\mathrm{G}}\boldsymbol{\beta}^{\mathrm{G}}+\mathbf{Z}^{\mathrm{G}}\mathbf{u}^{\mathrm{G}} = \mathbf{X}^{\mathrm{R}}\boldsymbol{\beta}^{\mathrm{R}}+\mathbf{Z}^{\mathrm{R}}\mathbf{u}^{\mathrm{R}}+\mathbf{X}^{\mathrm{G}}\boldsymbol{\beta}^{\mathrm{G}}+\sum_{\ell=1}^{L}\mathbf{Z}_{\ell}^{\mathrm{G}}\mathbf{u}_{\ell}^{\mathrm{G}},$$

where

$$\mathbf{X}^{\mathrm{R}} \equiv \begin{bmatrix} \mathbf{X}_1^{\mathrm{R}} \\ \vdots \\ \mathbf{X}_m^{\mathrm{R}} \end{bmatrix}, \qquad \mathbf{Z}^{\mathrm{R}} \equiv \underset{1\le i\le m}{\mathrm{blockdiag}}(\mathbf{X}_i^{\mathrm{R}}), \qquad \mathbf{Z}^{\mathrm{G}} = [\mathbf{Z}_1^{\mathrm{G}} \, \ldots \, \mathbf{Z}_L^{\mathrm{G}}],$$

with

$$\mathrm{Cov}(\mathbf{u}^{\mathrm{R}}) = \underset{1\le i\le m}{\mathrm{blockdiag}}(\boldsymbol{\Sigma}^{\mathrm{R}}) \qquad \text{and} \qquad \mathrm{Cov}(\mathbf{u}^{\mathrm{G}}) = \underset{1\le \ell\le L}{\mathrm{blockdiag}}(\sigma_{u\ell}^2\mathbf{I}).$$

The design matrices and parameter vectors with superscript “R” correspond to random intercepts and slopes, as typically used for repeated measures of data on m groups with sample sizes $n_1,\ldots,n_m$. For $1\le i\le m$, $\mathbf{X}_i^{\mathrm{R}}$ is a $n_i\times d^{\mathrm{R}}$ matrix for the random design corresponding to the ith group, $\boldsymbol{\Sigma}^{\mathrm{R}}$ is an unstructured $d^{\mathrm{R}}\times d^{\mathrm{R}}$ covariance matrix. The design matrices and parameter vectors with superscript “G” correspond to general design matrix structure. This allows, for example, the incorporation of non-linear covariate effects via the mixed model representation of penalised splines (Wand 2003).

In longitudinal data analysis the $\mathbf{Z}^{\mathrm{G}}\mathbf{u}^{\mathrm{G}}$ component is usually absent and d^{R} is a small integer, typically in the range 1–3. For example, a single covariate Poisson mixed model with random intercept:

$$y_{ij}\ \text{i.i.d. Poisson}\left\{\exp(\beta_0+\beta_1x_{ij}+U_i)\right\}, \quad U_i\ \text{i.i.d.}\ N(0,\sigma_U^2), \quad 1\le j\le n_i,\ 1\le i\le m, \tag{5}$$

corresponds to $d^{\mathrm{R}}=1$ and $\mathbf{X}_i^{\mathrm{R}}$ equalling the $n_i\times 1$ vector of ones. The log-likelihood (2) then reduces to

$$\ell(\beta_0,\beta_1,\sigma_U^2) = \sum_{i=1}^{m}\sum_{j=1}^{n_i}\{y_{ij}(\beta_0+\beta_1x_{ij})-\log(y_{ij}!)\} - \frac{m}{2}\log(2\pi\sigma_U^2)$$
$$+\sum_{i=1}^{m}\log\int_{-\infty}^{\infty}\exp\left(\sum_{j=1}^{n_i}y_{ij}U_i - e^{\beta_0+\beta_1x_{ij}+U_i} - \frac{U_i^2}{2\sigma_U^2}\right)\mathrm{d}U_i, \tag{6}$$

and the remaining intractable integrals are one-dimensional. However, if (5) is extended to the semiparametric model

$$y_{ij}\ \text{i.i.d. Poisson}\{\exp(\beta_0+\beta_1x_{ij}+f(s_{ij})+U_i)\},$$

where $f(s) = \beta_2 s + \sum_{k=1}^{K} u_k(s - \kappa_k)_+$ is a K-knot penalised spline model for f with u_k i.i.d. $N(0, \sigma_u^2)$ then the log-likelihood is

$$\begin{aligned}
&\ell\left(\beta_0, \beta_1, \beta_2, \sigma_U^2, \sigma_u^2\right) \\
&\quad = \sum_{i=1}^{m} \sum_{j=1}^{n_i} \{y_{ij}(\beta_0 + \beta_1 x_{ij} + \beta_2 s_{ij}) - \log(y_{ij}!)\} - \frac{m}{2}\log(2\pi\sigma_U^2) - \frac{K}{2}\log(2\pi\sigma_u^2) \\
&\qquad + \log \int_{\mathbb{R}^K} \left(\prod_{i=1}^{m} \int_{-\infty}^{\infty} \exp\left\{ \sum_{j=1}^{n_i} y_{ij} U_i - e^{\beta_0 + \beta_1 x_{ij} + \beta_2 s_{ij} + U_i + \sum_{k=1}^{K} u_k (s_{ij} - \kappa_k)_+} - \frac{U_i^2}{2\sigma_U^2} \right\} \mathrm{d}U_i \right) \\
&\qquad \times \exp\left\{ \sum_{i=1}^{m} \sum_{j=1}^{n_i} \sum_{k=1}^{K} y_{ij} u_k (s_{ij} - \kappa_k)_+ - \sum_{k=1}^{K} \frac{u_k^2}{2\sigma_u^2} \right\} \mathrm{d}u_1 \cdots \mathrm{d}u_K.
\end{aligned}$$

The dominating intractable integral is over $\mathbb{R}^K$.

Starting mainly with the Laplace approximation papers of Breslow and Clayton (1993) and Wolfinger and O'Connell (1993), there has been a great deal of research and software development for GLMM. Chapter 10 of McCulloch and Searle (2000) surveys the earlier literature. Zhao et al. (2006) reflects contemporary work in GLMM. The `SAS` procedure `PROC NLMIXED` (SAS Institute, Inc. 2007) uses quadrature to evaluate low-dimensional integrals such as those arising in (6). Markov Chain Monte Carlo packages based on the `BUGS` sampling engine (BUGS Project, 2007) have been successfully used in Bayesian GLMMs involving higher-dimensional integrals; see for example Crainiceanu et al. (2005), Gurrin et al. (2005) and Zhao et al. (2006).

2.2 Generalised Linear Models for Longitudinal Data with Serial Dependence

Models of this type are emerging more frequently in the longitudinal data analysis literature—for example see Diggle et al. (2002). Here

$$\mathbf{W} = \mathbf{I}, \quad \mathbf{w} = \mathbf{Z}^{\mathrm{R}}\mathbf{u}^{\mathrm{R}} + \boldsymbol{\alpha},$$

where $\mathbf{Z}^{\mathrm{R}}$ and $\mathbf{u}^{\mathrm{R}}$ represent the random effects components and $\boldsymbol{\alpha}^{\top} = (\boldsymbol{\alpha}_1^{\top}, \ldots, \boldsymbol{\alpha}_m^{\top})$ where

$$\boldsymbol{\alpha}_i \text{ i.i.d. } N(\mathbf{0}, \boldsymbol{\Gamma}_i), \quad i = 1, \ldots, m,$$

allow for serial dependence within the time ordered repeated measures on the m cases. The covariance matrix $\boldsymbol{\Gamma}_i = \boldsymbol{\Gamma}_i(\boldsymbol{\lambda})$ is a $n_i \times n_i$ Toeplitz matrix corresponding to an assumption of stationary serial dependence and is specified using a finite vector of parameters $\boldsymbol{\lambda}$ which is the same for all values of i. Let

$$\boldsymbol{\Gamma} = \underset{1 \le i \le m}{\mathrm{blockdiag}}(\boldsymbol{\Gamma}_i).$$

Computation of the log-likelihood for this model requires computation of m integrals with maximum dimension $d = \max_{1 \le i \le m} n_i$. Diggle et al. (2002) discuss computational approaches but do not present examples in which both $\mathbf{Z}^{\mathrm{R}}\mathbf{u}^{\mathrm{R}}$ and $\boldsymbol{\alpha}$ are present in the model simultaneously.

The class of models just considered may also be appropriate for the situation in which m time series of equal length $n_i \equiv n$ are observed. For example in Bernat et al. (2004) the impact of reduction of legal blood alcohol concentration from 0.1 to 0.08 on single vehicle night time fatalities was investigated in the USA. In that study monthly time series of fatalities in $m = 19$ states each of length $n_i = 72$ were analysed using a Poisson mixed model regression with random effects for before and after the change in legal blood alcohol levels. In some of the states significant autocorrelation existed. Other examples that we have encountered include the analysis of monthly suicide counts over a 30 year period in all of the states. Here $n_i \equiv 360$ and $m \approx 50$. These situations differ somewhat from the longitudinal data situation in that m is typically smaller and the n_i are typically much larger than in the longitudinal setting.

2.3 Generalised Linear Models for Time Series Regression

For this paper we will investigate the calculation of a single integral corresponding to a single series in Section 2.2. These single time series models have been extensively studied, in their own right, in recent years. An understanding of the calculation of those integrals required for computing the likelihood in the single series setting will also be necessary for the multiple time series and longitudinal models of Section 2.2. They correspond to a single ($m = 1$ in the previous model) time series $Y_1, \ldots, Y_T$ of observations which, conditional on an observed sequence of vectors of regressors $\{x_t\}$, and an unobserved stationary latent process $\{w_t\}$, are observations from an exponential family model or closely related family such as negative binomial. Throughout this paper we assume that the latent process is the autoregression of degree p, referred to as the AR(p) model henceforth,

$$w_t = \phi_1 w_{t-1} + \cdots + \phi_p w_{t-p} + \eta_t,$$

where η_t i.i.d. $N(0, \sigma^2)$. Here $\mathbf{w}$ is a $d = T$ dimensional vector with multivariate normal distribution specified to have mean vector $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma} = \mathrm{Cov}(\mathbf{w})$ with (s, t)th element given by the autocovariance at lag $|s - t|$ for the above AR(p) process. Let $\mathbf{V} = \mathbf{\Sigma}^{-1}$ and denote the parameters specifying the covariances as $\theta = (\phi_1, \ldots, \phi_p, \sigma^2)$. Then the log-likelihood of the observed data is as given above with $\mathbf{W} = \mathbf{I}$. Note that the integral required to be calculated is d-dimensional with $d = T$. It is not uncommon to have the length T of the time series in the thousands for realistic applications that arise in public health or financial econometrics.

Various simulation based methods currently available for computing the required integral are reviewed in Davis and Rodriguez-Yam (2005). Techniques reviewed include the various computationally intensive methods of importance sampling, Monte Carlo EM, and Monte Carlo Newton Raphson. Davis and Rodriguez-Yam (2005) introduce an approximation, based on the second order Taylor series expansion, to the conditional density of w_t given the count observations y_t and exploit the computationally efficient innovations algorithm to evaluate the resulting approximate likelihood. They also use this approximation to develop an approximate importance sampling technique which is computationally much faster than importance sampling. Application to a stochastic volatility model for 946 daily Pound-Dollar exchange rates is presented to illustrate the use of their methods beyond the exponential family setting discussed here.

3 Elements of Quasi-Monte Carlo Methods

In this section we provide a brief introduction to *quasi-Monte Carlo* (*QMC*) *methods*, with a focus on recent developments in *lattice rules*. We start our discussion with the Monte Carlo method, to which the QMC methods are related in a natural way.

3.1 The Monte Carlo Method and Importance Sampling

Consider an integral

$$\int_{\mathbb{R}^d} g(\mathbf{w})\, p(\mathbf{w})\, \mathrm{d}\mathbf{w}, \tag{7}$$

where $\mathbf{w} = (w_1, \ldots, w_d)^\top$ is a d-dimensional vector and p is some multivariate probability density function. Integrals of the form (7) often arise from multivariate expected values, and in many cases p is the multivariate normal density

$$(2\pi)^{-d/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left\{ -\frac{1}{2} (\mathbf{w} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{w} - \boldsymbol{\mu}) \right\} \tag{8}$$

with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The integral in the log-likelihood (2) is one example; many finance problems also have integrals of this form. The classical Monte Carlo (MC) method approximates (7) by an average of function values of g

$$\frac{1}{N} \sum_{i=1}^{N} g(\boldsymbol{\xi}_i),$$

where $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N \in \mathbb{R}^d$ are N independent random samples drawn from the density p. The expected error is of order $\mathcal{O}(N^{-1/2})$ and the efficiency depends on how easy it is to sample from the density p and how well p captures the features of the integrand $g(\mathbf{w})p(\mathbf{w})$. More precisely, the root mean square expected error is $\sigma(g)/\sqrt{N}$, where

$$\sigma^2(g) := \int_{\mathbb{R}^d} g^2(\mathbf{w})\, p(\mathbf{w})\, \mathrm{d}\mathbf{w} - \left(\int_{\mathbb{R}^d} g(\mathbf{w})\, p(\mathbf{w})\, \mathrm{d}\mathbf{w} \right)^2,$$

which we shall refer to as the *variance* of g.

For a more general integral

$$\int_{\mathbb{R}^d} f(\mathbf{w})\, \mathrm{d}\mathbf{w}, \tag{9}$$

the integrand f may not have the obvious form of some function times a density, or perhaps it has such a form but the density does not reflect the features of f. In these situations it is up to the user to choose an appropriate density p and rewrite (9) in the form of (7), with $g(\mathbf{w}) = f(\mathbf{w})/p(\mathbf{w})$. If an attempt is made to minimise $\sigma(g)$ then this technique is known as *importance sampling* and p is referred to as the *importance sampling density*. In practice, it is convenient for sampling if the sampling density p

is a product of univariate probability density functions. For example, if p is a normal density (8), then a change of variables $\mathbf{z} = \mathbf{A}^{-1}(\mathbf{w} - \boldsymbol{\mu})$ will achieve this, where $\mathbf{A}$ is a factorisation of $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^\top$. We shall assume for the remainder of this subsection that $p(\mathbf{w}) := \prod_{j=1}^{d} \psi(w_j)$, where ψ is a univariate probability density.

Importance sampling can be thought of as a change of variables. Let $\Psi : \mathbb{R} \to [0, 1]$ denote the cumulative distribution function of ψ, that is, $\Psi(w) = \int_{-\infty}^{w} \psi(t)\,\mathrm{d}t$, and let Ψ^{-1} denote its inverse. Using the substitution $\mathbf{z} = \Psi(\mathbf{w}) := (\Psi(w_1), \ldots, \Psi(w_d))^\top$, the integral (7) can be transformed into an integral over the unit cube

$$\int_{[0,1]^d} g\left(\Psi^{-1}(\mathbf{z})\right) \mathrm{d}\mathbf{z},$$

where $\Psi^{-1}(\mathbf{z}) := (\Psi^{-1}(z_1), \ldots, \Psi^{-1}(z_d))^\top$. The Monte Carlo method in effect approximates this integral by

$$\frac{1}{N} \sum_{i=1}^{N} g\left(\Psi^{-1}(\boldsymbol{\xi}_i)\right),$$

where now $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N \in [0, 1]^d$ are i.i.d. *uniform* random samples drawn from $[0, 1]^d$. In other words, the points are first drawn from a uniform distribution in $[0, 1]^d$ and then mapped into $\mathbb{R}^d$ using Ψ^{-1}. This is equivalent to the earlier discussion, where the points are generated directly from the density p. For the present approach to work in practice, we must have a way to evaluate Ψ^{-1}. Thus this approach is rarely used explicitly by people following the Monte Carlo strategy.

Now we come to the point where quasi-Monte Carlo methods depart from the Monte Carlo method. QMC methods take the same form as the Monte Carlo method in the unit cube $[0, 1]^d$, but instead of sampling the points randomly from a uniform distribution, the points are chosen deterministically in a clever way so that a rate of convergence of order $\mathcal{O}(N^{-1}(\log N)^d)$ or better is achieved. Since all QMC methods are defined in the unit cube, to apply QMC methods to a particular integral, one must first transform the integral into the unit cube. From our earlier discussion, it should be clear that importance sampling, with an appropriately chosen density, will give us the transformation we need. For the QMC strategy to succeed it is crucial that we can evaluate Ψ^{-1}, either analytically or numerically.

3.2 Low-discrepancy Point Sets

Assume now that the original integral in $\mathbb{R}^d$ has been transformed into an integral over the unit cube $[0, 1]^d$. We consider integrals of the form

$$If := \int_{[0,1]^d} f(\mathbf{z})\,\mathrm{d}\mathbf{z}.$$

Then QMC methods approximate such integrals by

$$Q_N f := \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{\xi}_i),$$

where $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N$ are points from the unit cube $[0, 1]^d$ chosen deterministically to be "more uniform than random". While the root mean square MC error is $\sigma(f)/\sqrt{N}$ where $\sigma^2(f) = If^2 - (If)^2$, QMC error bounds typically take the form

$$|If - Q_N f| \leq D(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N)\, V(f),$$

where $D(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N)$ is some *discrepancy*, which measures the quality (the "uniformity") of the points, and $V(f)$ is a measure of the variation of f. For example, the classical Koksma–Hlawka inequality has QMC error bounded by the so-called *star discrepancy* of the point set and the variation of f in the sense of Hardy and Krause (e.g. Niederreiter 1992). Error bounds of this form separate the dependence on the point sets from the dependence on the integrand. In general we do not have control over the integrand, and so we choose QMC points that make the discrepancy as small as possible. This is the principle behind all constructions of QMC methods.

One may ask the question: if the aim is to achieve high uniformity, why not just divide the unit interval in each coordinate direction into $n - 1$ equal segments and take the n^d grid points as our integration points? Clearly this is infeasible even for $n = 2$ if d is say 100. Another answer is that the projection of these $N = n^d$ points collapse down to just n distinct values in each coordinate direction, and thus for a function that depends only on a single component of $\mathbf{z}$ we receive only the benefit of having n points, yet have n^d costly function evaluations. More generally, one can apply a one-dimensional integration rule such as the Simpson rule in each coordinate direction to form what is called a *product* rule. However, even if the chosen one-dimensional rule has error of order $\mathcal{O}(n^{-r})$, the error expressed in terms of N is only of order $\mathcal{O}(N^{-r/d})$. In other words, the cost (in terms of function evaluations) for a given level of accuracy increases exponentially in d. It is this *curse of dimensionality* that makes product rules useless for high dimensional integrals.

QMC point sets with discrepancy of order $\mathcal{O}(N^{-1}(\log N)^d)$ or better are collectively known as *low-discrepancy point sets*. This includes, for example, the well known Sobol′ sequence and Niederreiter sequence. These are "sequences" because additional points can be added at any time. Their "extensibility" in N makes them very attractive in practice: if at any stage we decide to increase the number of points in our QMC approximation, we only need to evaluate the function at the additional points. However, point sets which are not extensible in N often have smaller discrepancies.

Digital nets and *lattice rules* are the two foci in recent QMC research. They represent two different strategies for achieving high uniformity of the points in the unit cube. For a survey of earlier work, see Niederreiter (1992) and Sloan and Joe (1994). Note that the definition of digital nets and lattice rules do not explicitly forbid product grids. Instead product grids are excluded by the design concepts based on minimising various forms of discrepancy measures.

The concept behind digital nets is based on having the right number of points in various sub-divisions of the unit cube. Figure 2a shows the first 64 points of the 2-dimensional Sobol′ sequence, which is an example of a digital $(0, 6, 2)$-net in base 2. If we divide the unit square into 64 strips with width $1/64$ in either direction then each strip contains exactly one point (with points on the boundary counting towards the next strip). Similarly if we divide the unit square into 64 squares, we get exactly one point in each square. In fact as long as the unit square is divided into 64 rectangles of the same shape and size, each rectangle will include exactly one point.

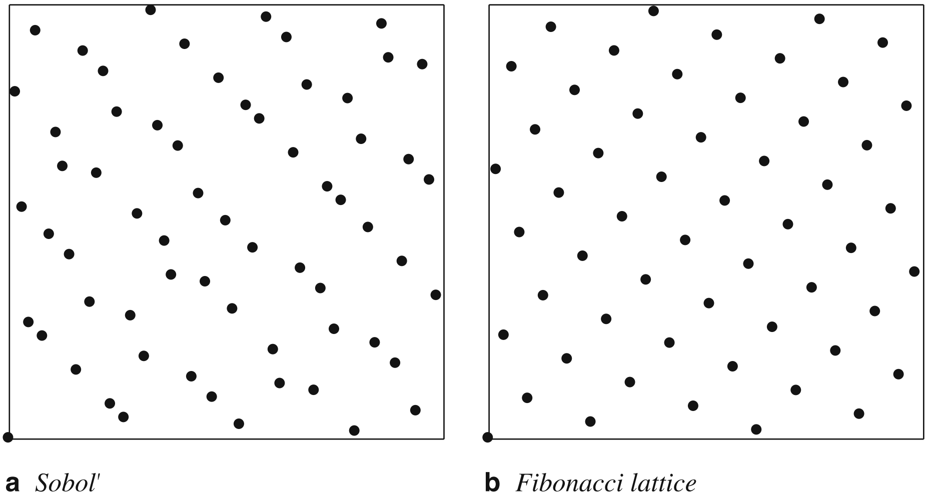

Fig. 2 Digital net versus lattice rule

The quality of a "(t, m, s)-net in base b" rests upon its t-value: the smaller t is, the finer the sub-divisions can be, while preserving the uniformity described above. (The ideal case above corresponds to $t = 0$. In a general (t, m, s)-net in base b there are b^m points in total, with b^t points in each sub-division. The dimension is typically denoted by s rather than d.) See Niederreiter (2005) for a survey of recent work on the construction of (t, m, s)-net.

Lattice rules have a different kind of uniformity: the points of a lattice rule form a group under the operation of addition modulo the integers. One way to visualise a lattice point set is to think of a sheared product grid where the axes have been stretched and rotated under certain constraints. The oldest and simplest kind of lattice rules, now known as *rank-1 lattice rules*, are uniquely specified by the choice of a *generating vector* $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_d)$, which is an integer vector having no factor in common with N. More precisely, the lattice points are given by $\boldsymbol{\xi}_i = \text{frac}(i\boldsymbol{\eta}/N)$ for $i = 1, \ldots, N$, where frac($\mathbf{x}$) is the vector obtained by replacing each component of the vector $\mathbf{x}$ by its fractional part. Figure 2b shows a "Fibonacci" lattice rule with $N = 55$ and $\boldsymbol{\eta} = (1, 34)$.

Deterministic methods have their disadvantages. Although the error is fully deterministic rather than probabilistic, in general the discrepancy bound tends to be far too pessimistic. In other words, QMC methods lack a practical error estimate, whereas the Monte Carlo expected error can easily be obtained by estimating the variance of the function. To overcome this problem, it is recommended that "randomised" QMC methods should be used in practice.

"Shifting" is the simplest form of randomisation. The idea is to move all the points in the same direction by the same amount, and if any point falls outside the unit cube then it is "wrapped" back into the cube from the opposite side. A short description on how to use *randomly-shifted* QMC methods to provide error estimation is given in the Appendix. Another popular but more complicated randomisation method is known as "scrambling". For a survey of randomisation techniques, see Hickernell and Hong (2002), L'Ecuyer and Lemieux (2002), and the references therein. Note that *shifting preserves the lattice structure* while *scrambling preserves the net structure*: a shifted net may no longer remain a net while a scrambled lattice could cease to be a lattice.

Low-discrepancy sequences such as the Sobol′ sequence have been widely used for many years by practitioners across various disciplines. In particular, there have been several documented successes for high-dimensional financial derivative calculations such as mortgage-backed securities and option pricing, see e.g. Paskov and Traub (1995). However, modern analysis and design of digital nets and lattice rules have so far been confined to academic exercises. In many cases these analyses rely on artificial assumptions which are not satisfied in real world problems. Furthermore, interdisciplinary collaborations are needed to properly test these newly developed methods.

Our experiments in this paper will focus on the use of randomly-shifted lattice rules, and as a comparison we consider also the Sobol′ sequence which is a classical example of digital nets. Unlike most low-discrepancy sequences obtained following some generic scheme, the design of lattice rules requires educated tuning. This might be viewed by some as a disadvantage but others will see it as an opportunity: we have the option and scope to tailor the lattice rule to the integrand. Thus before we can discuss the choice of a lattice rule, we must consider the nature of the integrand.

3.3 ANOVA Decomposition and Effective Dimension

It has been observed that the integrands in many practical problems have low *effective dimensions*. To see what this means, consider for example the simple integrand $f(z_1, z_2, z_3, z_4) = z_1 + \cos(z_2 z_3)$ whose nominal dimension is 4. One could say that the effective dimension is 3 because the integrand depends only on the first three variables. On the other hand, any sensible person would treat this problem as the sum of a one-dimensional integral and a two-dimensional integral, thus concluding that the effective dimension is only 2.

More generally, every d-dimensional function f can be decomposed as a sum of 2^d terms

$$f(\mathbf{z}) = \sum_{\mathfrak{u}\subseteq\{1,\ldots,d\}} f_{\mathfrak{u}}(\mathbf{z}_{\mathfrak{u}}), \tag{10}$$

where $\mathbf{z}_{\mathfrak{u}}$ denotes the set of variables $\{z_j : j \in \mathfrak{u}\}$, and each term $f_{\mathfrak{u}}$ depends only on the set of variables $\mathbf{z}_{\mathfrak{u}}$. This is known as the *ANOVA* (*analysis of variance*) *decomposition* if we impose the condition that $\int_0^1 f_{\mathfrak{u}}(\mathbf{z}_{\mathfrak{u}})\,\mathrm{d}z_j = 0$ for all $j \in \mathfrak{u}$. It can be shown that the ANOVA terms are orthogonal, i.e. $\int_{[0,1]^d} f_{\mathfrak{u}}(\mathbf{z}_{\mathfrak{u}}) f_v(\mathbf{z}_v)\,\mathrm{d}\mathbf{z} = 0$ for all $\mathfrak{u} \neq v$, and they can be expressed recursively by $f_{\mathfrak{u}}(\mathbf{z}_{\mathfrak{u}}) = \int_{[0,1]^{d-|\mathfrak{u}|}} f(\mathbf{z})\,\mathrm{d}\mathbf{z}_{-\mathfrak{u}} - \sum_{v\subset\mathfrak{u}} f_v(\mathbf{z}_v)$, where $\mathbf{z}_{-\mathfrak{u}} := \{z_j : j \notin \mathfrak{u}\}$. Moreover, we have

$$\sigma^2(f) = \sum_{\mathfrak{u}\subseteq\{1,\ldots,d\}} \sigma^2(f_{\mathfrak{u}}), \tag{11}$$

that is, the variance of f is the sum of the variances of the ANOVA terms.

We say that $f_{\mathfrak{u}}$ describes the "interaction" between the variables in $\mathbf{z}_{\mathfrak{u}}$, and we refer to the terms $f_{\mathfrak{u}}$ with $|\mathfrak{u}| = \ell$ collectively as the order-ℓ terms. For some functions it may be that only the terms involving say the first ten variables are important; or on the other hand it may be that all variables are equally important but the higher order interactions are negligible compared with the lower order ones. In both cases these functions are said to have low effective dimensions. Following the definitions in Caflisch et al. (1997) and Liu and Owen (2006):

- The *truncation dimension* of f is the smallest integer d_T such that
$$\sum_{\mathrm{u}\subseteq\{1,\ldots,d_T\}} \sigma^2(f_{\mathrm{u}}) \;\ge\; 0.99\,\sigma^2(f).$$
- The *superposition dimension* of f is the smallest integer d_S such that
$$\sum_{|\mathrm{u}|\le d_S} \sigma^2(f_{\mathrm{u}}) \;\ge\; 0.99\,\sigma^2(f).$$
- The *mean dimension* of f is
$$d_M := \sum_{\ell=1}^{d} \left(\ell \cdot \frac{1}{\sigma^2(f)} \sum_{|\mathrm{u}|=\ell} \sigma^2(f_{\mathrm{u}}) \right).$$

Thus the truncation dimension corresponds roughly to the number of important variables, while the superposition dimension describes the highest order of significant interactions between variables. The mean dimension is the expected order of f, which has a similar interpretation to the superposition dimension.

Note that in general it is impossible to obtain a simple expression for f_{u}. Fortunately there are ways to compute the effective dimensions of f without having to know the ANOVA terms, see Wang and Fang (2003) and Liu and Owen (2006). A brief outline of the techniques is given in the Appendix. Note, however, that these techniques require the evaluation of some integrals involving f, and thus the problem of estimating the effective dimensions is at least as hard as the original integration problem.

3.4 Lattice Rules, Random Shifts, and Weights

Lattice rules were traditionally used to approximate integrals with periodic integrands. Their role for non-periodic integrands has been known only for the past half decade. Here we give a brief discussion on how to choose a good lattice rule. See Kuo and Sloan (2005) for a more detailed description of our methodology.

The first step in the modern analysis of lattice rules is to identify a function space H to which the integrand f belongs. In analogy with the ANOVA decomposition, we assume that f belongs to a *weighted Sobolev space* H, in which every function has the decomposition (10) and its variance satisfies (11). The space H contains functions with square-integrable mixed first derivatives, and the inner product of H is "weighted" following the notion first introduced by Sloan and Woźniakowski (1998). Without going into the full details, it suffices to say that we associate a weight, γ_{u}, with every set of variables $\mathbf{z}_{\mathrm{u}}$, which describes the level of interaction between the variables in $\mathbf{z}_{\mathrm{u}}$. The 2^d weights γ_{u} together model the relative importance between various sets of variables. Roughly speaking, a small γ_{u} means that the contribution of the ANOVA term f_{u} is small compared with the other ANOVA terms of f. We set $\gamma_\emptyset = 1$ to fix the scaling. The limiting case of $\gamma_{\mathrm{u}} = 0$ implies that $f_{\mathrm{u}} = 0$, or in other words, there is no interaction between the variables in $\mathbf{z}_{\mathrm{u}}$.

To have a function space that best describes our integrand f, we must choose the weights $\gamma_{\mathfrak{u}}$ according to the dimension structure of f. This corresponds to the variance allocation of f among its ANOVA terms. To limit our choices, we consider just the following settings (e.g. Sloan et al. 2004):

- With *product weights*, we assume that $\gamma_{\mathfrak{u}} := \prod_{j\in\mathfrak{u}} \gamma_{\{j\}}$, that is, the weight associated with the set of variables $\mathbf{z}_{\mathfrak{u}}$ is automatically assigned the product of the weights for each individual variable z_j in this set. This setting is useful when the truncation dimension is low.
- With *order-dependent weights*, we assume that $\gamma_{\mathfrak{u}} := \Gamma_{|\mathfrak{u}|}$, that is, the weight associated with the set of variables $\mathbf{z}_{\mathfrak{u}}$ depends only on the cardinality of $\mathfrak{u}$. This setting is useful when the superposition dimension is low but the truncation dimension is still high.
- We can also have *finite-order weights* by setting $\gamma_{\mathfrak{u}} = 0$ for all $|\mathfrak{u}| > q$, with some fixed number $q < d$. Many high-dimensional integrals in practice do appear to have a finite order of just 2 or 3.

Once we have determined the weights for our function space H, we study the *worst-case error* defined by

$$e_N(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N) := \sup_{\|f\|_H \leq 1} |If - Q_N f|.$$

From this definition, we see that $|If - Q_N f| \leq e_N(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n) \, \|f\|_H$. Thus the worst-case error can be thought of as a type of discrepancy measure. In general we do not have a computable expression for the worst-case error, except when the function space H is a reproducing kernel Hilbert space; this is indeed the case for the Sobolev space.

The worst-case error is then used as our search criterion in a *component-by-component* algorithm (e.g. Sloan et al. 2002) which constructs a good generating vector $\boldsymbol{\eta}$ for randomly-shifted lattice rules. In a nutshell, this is a greedy algorithm which selects the best choice for each component of $\boldsymbol{\eta}$, one at a time, while holding all previously chosen components fixed. There are many variants of this algorithm, including those which aim for good *embedded* or *extensible* lattice rules (e.g. Cools et al. 2006; Dick et al. 2007). With a clever implementation, these algorithms can produce, in a very short computational time, good lattice rules with thousands of dimensions and millions of points that achieve close to $\mathcal{O}(N^{-1})$ convergence. An outline of the most basic component-by-component algorithm is given in the Appendix.

Unfortunately, we shall see in the next section that the transformed integrands arising from the log-likelihood integrals do not belong to the weighted Sobolev spaces. This is because we get either functions which are unbounded near the boundary of the unit cube, or functions whose derivatives near the boundary are unbounded and not square-integrable. In other words, the nice theory discussed above cannot strictly be applied. (In fact, even though there are numerous empirical results showing that QMC methods work well for many finance problems, there has been no concrete theoretical justification because the transformed integrands do not lie in any of the theoretical function space settings.) Having said that, recently Kuo et al. (2006) introduced a new function space setting which includes the transformed integrands in the next section. Even though they were only able to prove a $\mathcal{O}(N^{-1/2})$ convergence for randomly-shifted lattice rules, this is still one big step towards having an applicable theory. Furthermore, it provides some sort of theoretical explanation for the common observation that "QMC always performs no worse than MC".

4 QMC for Log-likelihood Integrals

The log-likelihood (2) can be written as

$$\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}) \;=\; \log \int_{\mathbb{R}^d} \exp\{F(\mathbf{w})\}\,d\mathbf{w} - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \tfrac{d}{2}\log(2\pi) + \mathbf{1}^\top c(\mathbf{y}),$$

where

$$F(\mathbf{w}) \;=\; \mathbf{y}^\top(\mathbf{X}\boldsymbol{\beta} + \mathbf{W}\mathbf{w}) - \mathbf{1}^\top b(\mathbf{X}\boldsymbol{\beta} + \mathbf{W}\mathbf{w}) - \frac{1}{2}\mathbf{w}^\top \boldsymbol{\Sigma}^{-1}\mathbf{w}.$$

Assuming, for the moment, that the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$ are given and fixed, we now consider the approximation of $\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma})$.

4.1 Laplace Approximation

As already discussed briefly in Section 2, the Laplace method approximates $\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ using a multivariate normal approximation to the integrand $\exp\{F(\mathbf{w})\}$ via a second order Taylor series approximation of $F(\mathbf{w})$ about its stationary point.

More precisely, the gradient and Hessian of $F(\mathbf{w})$ are given by

$$\nabla F(\mathbf{w}) = \mathbf{W}^\top\mathbf{y} - \mathbf{W}^\top b'(\mathbf{X}\boldsymbol{\beta} + \mathbf{W}\mathbf{w}) - \boldsymbol{\Sigma}^{-1}\mathbf{w}$$
$$\nabla^2 F(\mathbf{w}) = -\mathbf{W}^\top \mathrm{diag}(b''(\mathbf{X}\boldsymbol{\beta} + \mathbf{W}\mathbf{w}))\mathbf{W} - \boldsymbol{\Sigma}^{-1}.$$

We choose $\mathbf{w}^*$ and $\boldsymbol{\Sigma}^*$ such that

$$\nabla F(\mathbf{w}^*) \;=\; \mathbf{0} \quad \text{and} \quad \boldsymbol{\Sigma}^* \;=\; \left\{-\nabla^2 F(\mathbf{w}^*)\right\}^{-1},$$

and we approximate $F(\mathbf{w})$ by

$$\begin{aligned} F(\mathbf{w}) &\approx\; F(\mathbf{w}^*) + \nabla F(\mathbf{w}^*)(\mathbf{w} - \mathbf{w}^*) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^\top \nabla^2 F(\mathbf{w}^*)(\mathbf{w} - \mathbf{w}^*) \\ &=\; F(\mathbf{w}^*) - \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^\top \boldsymbol{\Sigma}^{*-1}(\mathbf{w} - \mathbf{w}^*). \end{aligned}$$

Thus

$$\int_{\mathbb{R}^d} \exp\{F(\mathbf{w})\}\,d\mathbf{w} \;\approx\; (2\pi)^{d/2}|\boldsymbol{\Sigma}^*|^{1/2}\exp\{F(\mathbf{w}^*)\},$$

which leads to

$$\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}) \overset{\text{Laplace}}{\approx} F(\mathbf{w}^*) + \frac{1}{2}\log|\boldsymbol{\Sigma}^*| - \frac{1}{2}\log|\boldsymbol{\Sigma}| + \mathbf{1}^\top c(\mathbf{y}).$$

4.2 QMC Approximation

We now approximate the integral in $\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ using QMC methods. To do this we need to transform this integral into the unit cube. As we have explained in Section 3, this is equivalent to importance sampling for the Monte Carlo method. The transformation plays a crucial role as it controls the feature and dimension structure of our transformed integrand. In other words, the transformation determines the difficulty of the integration problem over the unit cube.

The transformation process comprises two stages:

1. Translate and re-centre the mode, and rotate and re-scale the axes.
2. Map the integral into the unit cube.

The first stage helps to eliminate spiky integrands and integrands with support far away from the origin in $\mathbb{R}^d$. This can be achieved by a change of variables

$$\mathbf{v} = \mathbf{P}^{-1}(\mathbf{w} - \boldsymbol{\mu}),$$

with a suitably chosen centre $\boldsymbol{\mu}$ and an invertible matrix $\mathbf{P}$ so that the components of $\mathbf{v}$ are properly scaled with respect to each other. The second stage is to map every integration variable v_j from $\mathbb{R}$ into [0, 1] using the same cumulative distribution function, that is, we use the substitution

$$\mathbf{z} = \Psi(\mathbf{v}) := (\Psi(v_1), \ldots, \Psi(v_d))^\top,$$

where Ψ is the cumulative distribution function of a univariate probability density ψ. These two stages of the transformation process lead to

$$\begin{aligned}\int_{\mathbb{R}^d} \exp\{F(\mathbf{w})\}\, \mathrm{d}\mathbf{w} &= |\mathbf{P}| \int_{\mathbb{R}^d} \exp\{F(\boldsymbol{\mu} + \mathbf{P}\mathbf{v})\}\, \mathrm{d}\mathbf{v} \\ &= |\mathbf{P}| \int_{\mathbb{R}^d} \exp\{F(\boldsymbol{\mu} + \mathbf{P}\mathbf{v})\} \prod_{j=1}^{d} \frac{1}{\psi(v_j)} \times \prod_{j=1}^{d} \psi(v_j)\, \mathrm{d}\mathbf{v} \\ &= |\mathbf{P}| \int_{[0,1]^d} \exp\left\{F(\boldsymbol{\mu} + \mathbf{P}\Psi^{-1}(\mathbf{z}))\right\} \prod_{j=1}^{d} \frac{1}{\psi(\Psi^{-1}(z_j))}\, \mathrm{d}\mathbf{z},\end{aligned}$$

where $\Psi^{-1}(\mathbf{z}) := (\Psi^{-1}(z_1), \ldots, \Psi^{-1}(z_d))^\top$. Hence the transformed integrand is

$$f(\mathbf{z}) := |\mathbf{P}| \exp\left\{F(\boldsymbol{\mu} + \mathbf{P}\Psi^{-1}(\mathbf{z}))\right\} \prod_{j=1}^{d} \frac{1}{\psi(\Psi^{-1}(z_j))},$$

and a QMC rule with points $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N$ gives the approximation

$$\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}) \overset{\text{QMC}}{\approx} \log\left(\frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{\xi}_i)\right) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{d}{2}\log(2\pi) + \mathbf{1}^\top c(\mathbf{y}).$$

How should we choose the pair $\boldsymbol{\mu}$ and $\mathbf{P}$? Clearly we should re-centre our integrand based on the stationary point $\mathbf{w}^*$ of $F(\mathbf{w})$. On the other hand, the Laplace method provides a fairly good approximation locally around $\mathbf{w}^*$, and we can think of $\boldsymbol{\Sigma}^*$ from Section 4.1 as being chosen to match the curvature of the integrand at $\mathbf{w}^*$. This motivates the choice

$$\boldsymbol{\mu} = \mathbf{w}^* \qquad \text{and} \qquad \mathbf{P} = \mathbf{A}^*,$$

where $\mathbf{A}^*$ is a matrix satisfying

$$\boldsymbol{\Sigma}^* = \mathbf{A}^*\mathbf{A}^{*\top}.$$

An obvious choice is to take $\mathbf{A}^*$ to be the Cholesky factor of $\boldsymbol{\Sigma}^*$. Alternatively, $\mathbf{A}^*$ can be obtained from spectral decomposition as $(\sqrt{\lambda_1}\boldsymbol{\nu}_1, \ldots, \sqrt{\lambda_d}\boldsymbol{\nu}_d)$, where $\lambda_1 \geq \cdots \geq \lambda_d$ are the eigenvalues of $\boldsymbol{\Sigma}^*$ and $\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_d$ are the corresponding unit column eigenvectors. This second choice of decomposition is often associated with the term *principal component analysis* (PCA) in the QMC community even though all the eigenvalues and eigenvectors are used. We shall refer to this second choice of $\mathbf{A}^*$ as the PCA factor of $\boldsymbol{\Sigma}^*$.

Note that we could also take $\mathbf{P} = \mathbf{A}$, with $\mathbf{A}$ being the Cholesky or PCA factor of the covariance matrix $\boldsymbol{\Sigma}$ from the normal density already present in the integrand. However, exploratory calculations indicate that such a transformation leads to very poor results. The bottom line is that $\boldsymbol{\Sigma}$ simply does not capture the features of the integrand around $\mathbf{w}^*$.

Having chosen $\boldsymbol{\mu}$ and $\mathbf{P}$ for the first stage of the transformation process, we now choose a probability density ψ for the second stage.

Transformation 1 The most intuitive choice is to take ψ to be the standard normal density,

$$\psi(v) = (2\pi)^{-1/2} \exp\left(-\frac{1}{2}v^2\right).$$

The transformed integrand is then

$$f_1(\mathbf{z}) := (2\pi)^{d/2}|\boldsymbol{\Sigma}^*|^{1/2} \exp\left\{F(\mathbf{w}^* + \mathbf{A}^*\Psi^{-1}(\mathbf{z})) + \frac{1}{2}\Psi^{-1}(\mathbf{z})^\top\Psi^{-1}(\mathbf{z})\right\}.$$

Note that there is no closed form expression for Ψ^{-1}, although computational techniques based on rational approximation are well known.

Observe that f_1 is continuous on $(0, 1)^d$, but it could be unbounded near the boundaries of the unit cube. Using the definition of $\boldsymbol{\Sigma}^*$ we can write the exponent in f_1 as

$$\begin{aligned}\mathbf{y}^\top(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w}^*+\mathbf{W}\mathbf{A}^*\Psi^{-1}(\mathbf{z})) &- \mathbf{1}^\top b(\mathbf{X}\boldsymbol{\beta} + \mathbf{W}\mathbf{w}^* + \mathbf{W}\mathbf{A}^*\Psi^{-1}(\mathbf{z})) \\ &- \frac{1}{2}\mathbf{w}^{*\top}\boldsymbol{\Sigma}^{-1}\mathbf{w}^* - \mathbf{w}^{*\top}\boldsymbol{\Sigma}^{-1}\mathbf{A}^*\Psi^{-1}(\mathbf{z}) \\ &+ \frac{1}{2}(\mathbf{W}\mathbf{A}^*\Psi^{-1}(\mathbf{z}))^\top \operatorname{diag}(b''(\mathbf{X}\boldsymbol{\beta}+\mathbf{W}\mathbf{w}^*))(\mathbf{W}\mathbf{A}^*\Psi^{-1}(\mathbf{z})).\end{aligned}$$

When $b(x) = \log(1 + e^x)$, the positive quadratic term in the exponent dominates. Thus f_1 is unbounded at all boundaries of the cube. Consider now $b(x) = e^x$. Then the exponent is a balance between negative exponential and positive quadratic terms, and it could potentially approach ±∞. Thus in this case f_1 is unbounded at some boundaries of the cube. A more detailed analysis of the boundary behaviour for the latter case is given in the Appendix.

The transformation described above can lead to integrands which are unbounded at the boundaries of the unit cube. However, unbounded integrands do not belong to the Sobolev spaces discussed in Section 3, and therefore the existing theory on lattice rules cannot strictly be applied. Our aim is to find an alternative transformation such that the transformed integrand is bounded everywhere on $[0, 1]^d$.

Transformation 2 Instead of using the normal density to map the integral into the unit cube, we use the logistic density

$$\psi(v) = \frac{e^{v/\lambda}}{\lambda(1+e^{v/\lambda})^2}.$$

This density has a symmetric bell shape, but its tails only have exponential decay. The cumulative distribution function is

$$\Psi(v) = \frac{e^{v/\lambda}}{1+e^{v/\lambda}}, \quad \text{with} \quad \Psi^{-1}(z) = \lambda \log\left(\frac{z}{1-z}\right).$$

This inverse function is known as the *logit* function or the logistic map. The transformed integrand in this case is

$$f_2(\mathbf{z}) := \lambda^d |\mathbf{\Sigma}^*|^{1/2} \exp\left\{F(\mathbf{w}^* + \mathbf{A}^*\Psi^{-1}(\mathbf{z}))\right\} \prod_{j=1}^{d} \left(e^{-\Psi^{-1}(z_j)/\lambda} + 2 + e^{\Psi^{-1}(z_j)/\lambda}\right).$$

As before we consider the boundary behaviour of f_2. After grouping the product term into the exponential function, the exponent is

$$\mathbf{y}^\top\left(\mathbf{X}\boldsymbol{\beta} + \mathbf{W}(\mathbf{w}^* + \mathbf{A}^*\Psi^{-1}(\mathbf{z}))\right) - \mathbf{1}^\top b\left(\mathbf{X}\boldsymbol{\beta} + \mathbf{W}(\mathbf{w}^* + \mathbf{A}^*\Psi^{-1}(\mathbf{z}))\right)$$
$$-\frac{1}{2}(\mathbf{w}^*+\mathbf{A}^*\Psi^{-1}(\mathbf{z}))^\top \mathbf{\Sigma}^{-1}(\mathbf{w}^*+\mathbf{A}^*\Psi^{-1}(\mathbf{z})) + \sum_{j=1}^{d} \log(e^{-\Psi^{-1}(z_j)/\lambda} + 2 + e^{\Psi^{-1}(z_j)/\lambda}).$$

When $b(x) = \log(1+e^x)$, the negative quadratic term dominates the exponent and so f_2 is bounded everywhere on $[0, 1]^d$. If $b(x) = e^x$, then the exponent is dominated by negative exponential as well as negative quadratic terms, and it could only approach $-\infty$ and not $+\infty$. Thus in this case f_2 is also bounded everywhere on $[0, 1]^d$.

It is worth noting that even though the integrands are bounded, the derivatives can be huge. In practice it may actually be easier to handle an unbounded integrand with weak singularities rather than bounded integrands with huge boundary derivatives.

We could use any probability density for the second stage of the transformation. The standard choice is the normal density, while Evans and Swartz (1995) used the t density. The choice of the logistic density is new, and is attractive because of the closed form for the inverse cumulative distribution function. One other possible choice is to use a skewed normal density that has different standard deviations for positive and negative values. Since our integrand is skewed in some way, this transformation may prove to be advantageous.

5 Numerical Experiments

In our experiments here we consider two simplified problems in which we assumed that the parameters are given and fixed. The goal here is to find the best transformation to obtain good QMC approximations of the log-likelihood integrals. In the next section we will embed these integral calculations in an optimisation procedure to find the optimal parameter set.

Example 1 We consider a simple parameter driven Poisson state-space model where $\mathbf{w} = (w_1, \ldots, w_d)^\top$ is an autoregression of degree 1 (i.e., $w_j = \phi w_{j-1} + \eta_j$ with η_j i.i.d. $N(0, \sigma^2)$). Here $b(x) = e^x$, $c(x) = \log(1/x!)$, $\mathbf{W} = \mathbf{I}$, $\mathbf{X} = \mathbf{1}$, $\boldsymbol{\beta} = \beta$, and the covariance matrix is Toeplitz

$$\mathbf{\Sigma} = \frac{\sigma^2}{1-\phi^2}\begin{bmatrix} 1 & \phi & \phi^2 & \cdots & \phi^{d-2} & \phi^{d-1} \\ \phi & 1 & \phi & \cdots & \phi^{d-3} & \phi^{d-2} \\ \phi^2 & \phi & 1 & \cdots & \phi^{d-4} & \phi^{d-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi^{d-2} & \phi^{d-3} & \phi^{d-4} & \cdots & 1 & \phi \\ \phi^{d-1} & \phi^{d-2} & \phi^{d-3} & \cdots & \phi & 1 \end{bmatrix}, \quad \text{with} \quad |\mathbf{\Sigma}| = \frac{\sigma^{2d}}{1-\phi^2}.$$

The Cholesky factor of $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^\top$ and its inverse are known explicitly

$$\mathbf{A} = \sigma \begin{bmatrix} \frac{1}{\sqrt{1-\phi^2}} & 0 & 0 & \cdots & 0 \\ \frac{\phi}{\sqrt{1-\phi^2}} & 1 & 0 & \cdots & 0 \\ \frac{\phi^2}{\sqrt{1-\phi^2}} & \phi & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\phi^{d-1}}{\sqrt{1-\phi^2}} & \phi^{d-2} & \phi^{d-3} & \cdots & 1 \end{bmatrix} \text{ and } \mathbf{A}^{-1} = \frac{1}{\sigma}\begin{bmatrix} \sqrt{1-\phi^2} & 0 & 0 & \cdots & 0 & 0 \\ -\phi & 1 & 0 & \cdots & 0 & 0 \\ 0 & -\phi & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & -\phi & 1 \end{bmatrix}.$$

The inverse of $\mathbf{\Sigma}$ is tridiagonal

$$\mathbf{\Sigma}^{-1} = \frac{1}{\sigma^2}\begin{bmatrix} 1 & -\phi & 0 & \cdots & 0 & 0 \\ -\phi & 1+\phi^2 & -\phi & \cdots & 0 & 0 \\ 0 & -\phi & 1+\phi^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1+\phi^2 & -\phi \\ 0 & 0 & 0 & \cdots & -\phi & 1 \end{bmatrix}.$$

We have a data file containing count data $\mathbf{y} = (y_1, \ldots, y_d)^\top$ up to dimension $d = 200$. These data were simulated from the parameters $\beta = 0.7$, $\sigma^2 = 0.3$ and $\phi = 0.5$. Due to the time series structure, we can consider this integration problem with d taking any value up to 200. We shall consider $d = 25, 50, 100, 150, 200$.

For this example we have

$$F(\mathbf{w}) = \sum_{j=1}^{d} \left(\beta y_j + w_j y_j - e^{\beta} e^{w_j}\right) - \frac{1}{2}\mathbf{w}^\top \mathbf{\Sigma}^{-1}\mathbf{w},$$

$$\nabla F(\mathbf{w}) = \mathbf{y} - e^{\beta} e^{\mathbf{w}} - \mathbf{\Sigma}^{-1}\mathbf{w},$$

$$\nabla^2 F(\mathbf{w}) = -\mathrm{diag}(e^{\beta} e^{\mathbf{w}}) - \mathbf{\Sigma}^{-1}.$$

To obtain the stationary point $\mathbf{w}^*$ satisfying $\nabla F(\mathbf{w}^*) = \mathbf{0}$, we use the Newton iteration

$$\left(-\nabla^2 F\left(\mathbf{w}^{(k)}\right)\right)\left(\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\right) = \nabla F\left(\mathbf{w}^{(k)}\right),$$

starting at $\mathbf{w}^{(0)} = \mathbf{0}$. Each iteration can be solved by a Cholesky factorisation of $(-\nabla^2 F(\mathbf{w}^{(k)}))$ followed by forward and back substitutions, see the `LAPACK` function `DPOTRS` (Anderson et al. 1999). Once a satisfactory $\mathbf{w}^*$ is found, the matrix $\mathbf{\Sigma}^*$

discussed in Section 4.1 is given by $\mathbf{\Sigma}^* = \left(-\nabla^2 F(\mathbf{w}^*)\right)^{-1}$, where the inverse matrix can be obtained by making use of the Cholesky factorisation, see the `LAPACK` function `DPOTRI`.

Example 2 We consider a generalised linear mixed model which is a simple semi-parametric regression model with Poisson response. In this example $b(x) = e^x$, $c(x) = \log(1/x!)$, $\mathbf{w} = \mathbf{u} = (u_1, \ldots, u_d)^\top$, $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \{0, 1, 2, \ldots\}^n$, and

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{21} \\ 1 & x_{12} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}, \quad \mathbf{W} = \mathbf{Z} = \left[(x_{2i} - \kappa_j)_+\right]_{\substack{1\le i\le n \\ 1\le j\le d}},$$

where $x_{11}, \ldots, x_{1n} \in \{0, 1\}$, $x_{21}, \ldots, x_{2n} \in (0, 1)$, and $\kappa_1, \ldots, \kappa_d \in (0, 1)$ are fixed numbers ranging between the maximum and minimum of x_{2i}, known as knots. The covariance matrix is $\mathbf{\Sigma} = \sigma^2\mathbf{I}$.

We have a simulated data set of $\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}$, and numbers κ_j, for $d = 25$ and $n = 500$. (Note that the dimension for this example is fixed.) A good starting point for the parameters is given by $\beta_0 = 1$, $\beta_1 = 0.77$, $\beta_2 = 1.5$, and $\sigma^2 = 51$.

For this example we have

$$\begin{aligned} F(\mathbf{u}) &= \mathbf{y}^\top(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}) - \mathbf{1}^\top \exp(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}) - \mathbf{u}^\top\mathbf{u}/\left(2\sigma^2\right), \\ \nabla F(\mathbf{u}) &= \mathbf{Z}^\top(\mathbf{y} - \exp(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u})) - \mathbf{u}/\sigma^2, \\ \nabla^2 F(\mathbf{u}) &= -\mathbf{Z}^\top \mathrm{diag}(\exp(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}))\,\mathbf{Z} - \left(1/\sigma^2\right)\mathbf{I}. \end{aligned}$$

The stationary point $\mathbf{u}^*$ and the matrix $\mathbf{\Sigma}^*$ can be obtained in the same way as in Example 1.

Following Section 4.2, we consider four transformations: ψ is either the standard normal density or the logistic density with $\lambda = 0.6$, and $\mathbf{A}^*$ is either the Cholesky factor or the PCA factor of $\mathbf{\Sigma}^*$. For convenience we shall refer to these four transformation methods as “Normal + PCA”, “Normal + Cholesky”, “Logistic 0.6 + PCA”, and “Logistic 0.6 + Cholesky”. (The value of $\lambda = 0.6$ is chosen because it appears to give better approximations than other values of λ.)

5.1 Effective Dimension Estimation

To gain some insights to the dimension structure of these transformed integrands in the unit cube, we compute their effective dimensions. Figure 3 shows the estimated truncation dimension and mean dimension for both examples under the four transformations described above. (More details regarding how these numbers were obtained are given in Appendix.)

It is important to realise that these numbers can only be used a rough guide because the errors in these estimates can be huge. The superposition dimension estimates are the most unreliable of all and this is why we do not present the results here. (Past experiences indicate that the mean dimensions tend to be under-estimates of the superposition dimensions.) Having said that, we do get a clear picture of the dimension structure of these functions: the truncation dimension is essentially the

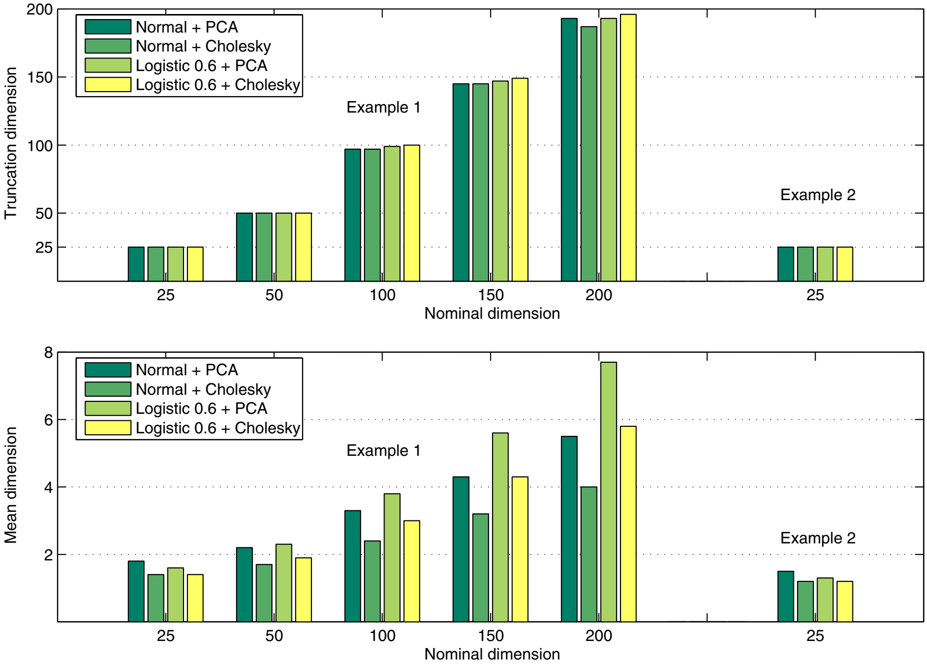

Fig. 3 Effective dimensions of transformed integrands

nominal dimension regardless of the transformation method. Unlike many option pricing problems in finance, the PCA decomposition does not help to reduce the truncation dimension for our examples here.

On the other hand, the transformation method appears to have some small effect on the mean dimension. Although no concrete conclusions can be drawn from these results, the PCA decomposition generally leads to higher mean dimension than the Cholesky decomposition does. As the nominal dimension increases, the mean dimension (and most likely the superposition dimension too) increases rapidly, indicating that these high-dimensional problems are truly high-dimensional. This is again contrary to many finance problems in which the mean dimension remains around 2 as the nominal dimension increases.

5.2 Two-dimensional Projections

Figures 4 and 5 give some two-dimensional projections of these transformed integrands selected to demonstrate the type of boundary behaviour that can occur.

In Fig. 4, the graph (a.1) corresponds to the transformed integrand obtained by using “Normal + PCA” in Example 1 with $d = 25$. It shows a function of the first and the fifth variables, with all other variables fixed at the value 0.5. In other words, it is a two-dimensional projection of the transformed integrand from the centre of the unit cube and parallel to the axes z_1 and z_5. The graph suggests that the function goes to infinity when either variable goes to 0, but it is otherwise fairly flat. The graphs (a.2) and (a.3) show other projections of the same transformed integrand in different

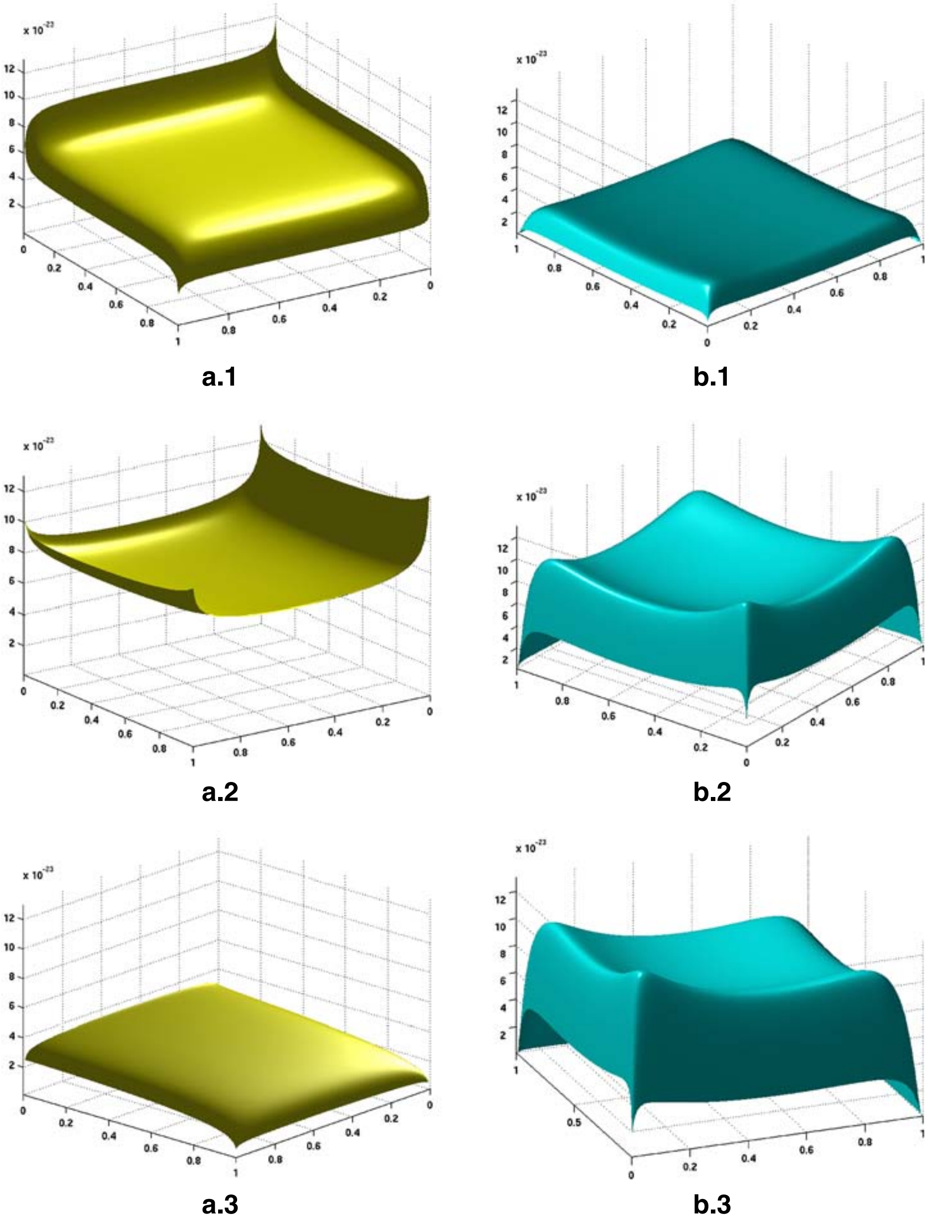

Fig. 4 Two-dimensional projections of transformed integrand obtained using (**a**) normal density and (**b**) logistic density with $\lambda = 0.6$

dimensions and anchored at different positions in the unit cube. In fact, for both Examples 1 and 2, the transformations based on the standard normal density lead to projections like those in (a), and the transformations based on the logistic density lead to projections like those in (b). This is regardless of whether PCA or Cholesky

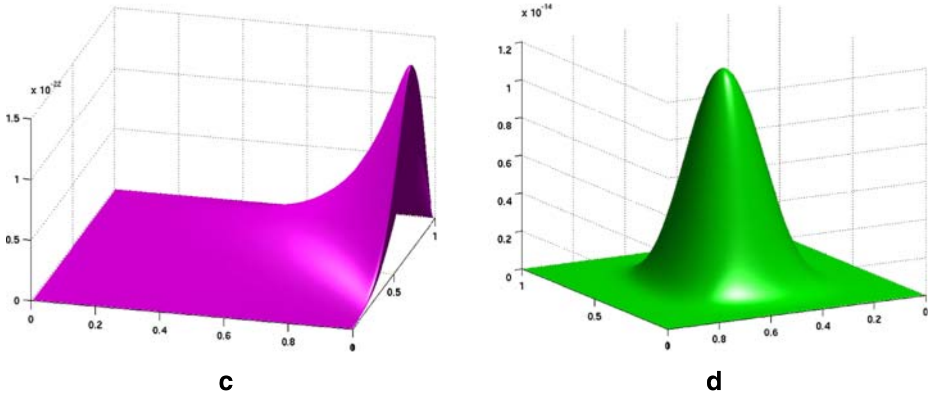

Fig. 5 Two-dimensional projections of transformed integrand obtained (**c**) without centring and rescaling and (**d**) with centring but without rescaling

decomposition is used. These features are as we have predicted: the use of standard normal density can result in unbounded integrands, and the use of logistic density always give bounded integrands but the derivatives near the boundary can be huge.

As a comparison, in Fig. 5 we present the projections of some badly transformed integrands: (c) shows the support of the integrand off to one corner and (d) shows a narrow spike at the centre. They are associated with transformed integrands obtained without centring, or with centring but without rescaling. We do not expect MC or QMC methods to do well for these badly transformed integrands (as it turned out, the results were way off). Further details about these figures are given in Appendix.

5.3 Integral Calculation

Now we compute the integral for each transformed integrand using the Monte Carlo method, the Sobol′ sequence, and four lattice rules constructed based on various choices of weights (see Appendix for details regarding these lattice rules). In each calculation, we use ten random shifts of N points, $N = 2^{15} = 32{,}768$ or $N = 2^{16} = 65{,}536$, and we estimate the standard errors. We then repeat this process 30 times and produce a comparison of the boxplots for the $\log_{10}$ relative integrals and $\log_{10}$ relative standard errors. The results for Example 1 with $d = 25$, Example 2 (which has $d = 25$), and Example 1 with $d = 200$ are presented in Figs. 6, 7, and 8, respectively. Note that since the boundary behaviour is the dominating feature of some transformed integrands, it is necessary that we use a good random number generator with extended precision (see Appendix for details).

More precisely, for each of the six integration rules and each of the four transformation methods, we compute

$$I_{s,k}(\text{rule, trans}) = \frac{1}{N}\sum_{i=1}^{N} f_{s,k,i}(\text{rule, trans})$$

for $s = 1, 2, \ldots, 30$ and $k = 1, 2, \ldots, 10$, where $f_{s,k,i}$ denotes the transformed integrand value at the kth random shift of the ith integration point during the sth repetition. For each repetition $s = 1, 2, \ldots, 30$, we compute the estimated integral and the corresponding standard error

$$\bar{I}_s(\text{rule, trans}) = \frac{1}{10}\sum_{k=1}^{10} I_{s,k}(\text{rule, trans}),$$

$$E_s(\text{rule, trans}) = \sqrt{\frac{1}{10 \times 9}\sum_{k=1}^{10}\left(I_{s,k}(\text{rule, trans}) - \bar{I}_s(\text{rule, trans})\right)^2}.$$

Each boxplot in the figures contains the logarithm (base 10) of a scaled version of 30 numbers, either

$$\log_{10}\left(\frac{\bar{I}_s(\text{rule, trans})}{\text{median}_{1\le s\le 30}\ \bar{I}_s(\text{MC, Normal + PCA})}\right),$$

or

$$\log_{10}\left(\frac{E_s(\text{rule, trans})}{\text{median}_{1\le s\le 30}\ \bar{I}_s(\text{MC, Normal + PCA})}\right).$$

In other words, all results have been scaled by the median of the MC estimated integral using "Normal + PCA". Thus the first boxplot in every figure should have its black dot exactly at 0.

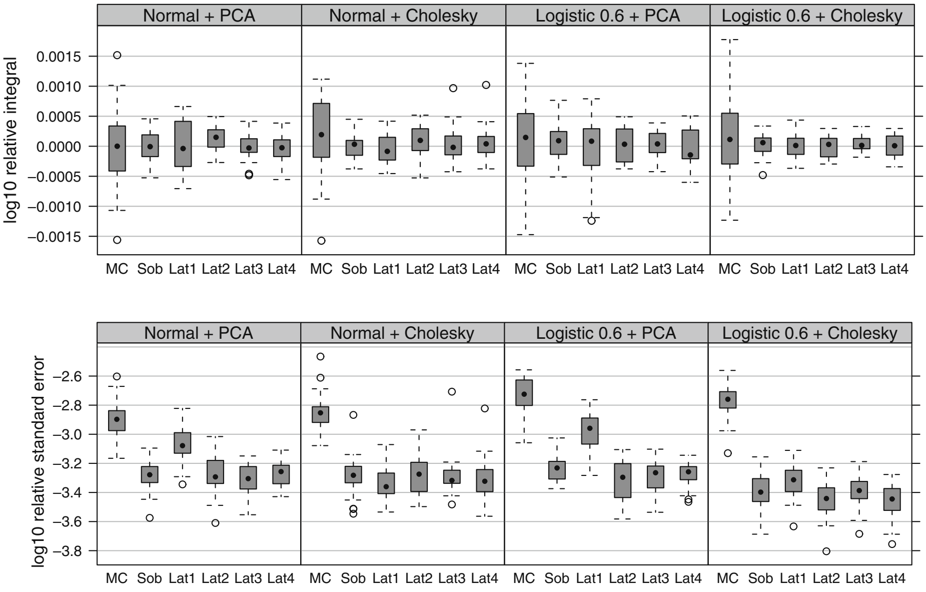

Fig. 6 Example 1 with $d = 25$. $\log_{10}$ relative integrals and $\log_{10}$ relative standard errors obtained from 30 repetitions based on ten random shifts of 32,768 points

Running on a typical 2006 desktop PC, the time required for computing one integral $\bar{I}_s(\text{rule}, \text{trans})$ in Figs. 6, 7, and 8 are roughly 8 sec, 30 sec, and 2.5 min, respectively.

Conclusion from Fig. 6

First we discuss the results in Fig. 6 for Example 1 with $d = 25$ and $N = 32,768$. At a glance it is clear that QMC dramatically outperforms MC: the integral estimates for both MC and QMC are unbiased, but the standard errors for MC are significantly larger than those for QMC.

For three of the four transformations, the lattice rule "Lat1" gives noticeably worse results than Sobol′ points and the other three lattice rules. This particular lattice rule is designed for integrands with only second-order interactions (assuming no higher order terms are present), and is clearly not suitable for our integrands here. It goes to show that the selection of weights in the design of lattice rules to match the dimension structure of the integrands indeed has some practical effects. Leaving this "bad" lattice rule aside, the other three lattice rules appear to give slightly better results than the Sobol′ points.

The best transformation for MC is "Normal + PCA". Note that the method of decomposition (i.e. PCA or Cholesky) should have no impact on the performance of MC theoretically. This is because the MC root mean square error depends on the total variance of the transformed integrand, which is invariant under different decompositions.

The best transformation for QMC is "Logistic 0.6 + Cholesky". Quantitatively, we take the average median of the relative standard errors for QMC (with "Lat1"

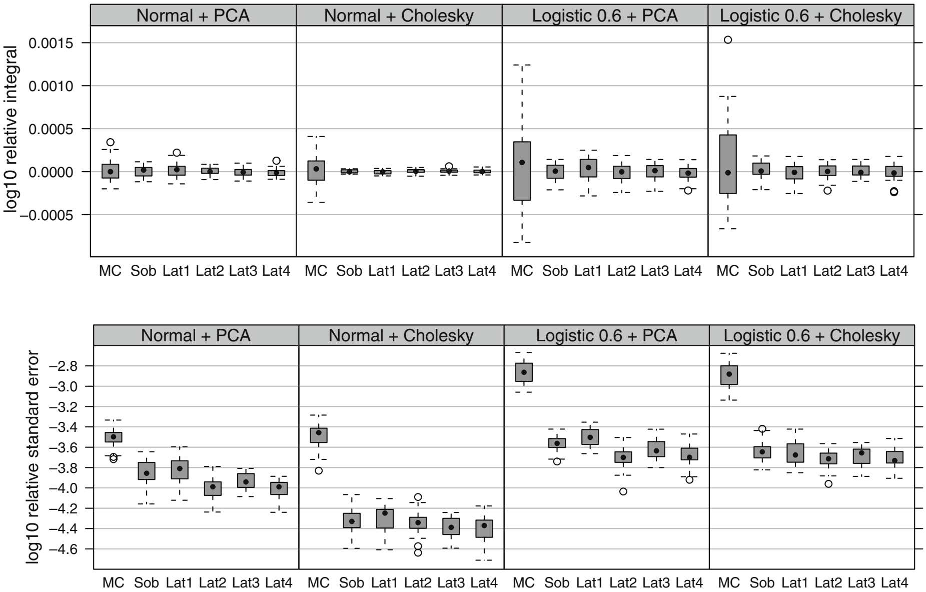

Fig. 7 Example 2 ($d = 25$). $\log_{10}$ relative integrals and $\log_{10}$ relative standard errors obtained from 30 repetitions based on ten random shifts of 32,768 points

excluded) under each transformation method and compare it with the median of the relative standard error for MC under its best transformation “Normal + PCA”. The improvement of QMC over MC in terms of the reduction in standard errors are 3.3 for “Logistic 0.6 + Cholesky”, 2.5 for “Normal + Cholesky”, 2.4 for “Normal + PCA”, and 2.3 for “Logistic 0.6 + PCA”. In other words, the MC standard errors can be more than three times the QMC standard errors.

Conclusion from Fig. 7

Figure 7 includes results for Example 2 (which has a fixed dimension $d = 25$) with $N = 32,768$. For this example, the superiority of QMC over MC stands out even more. It is also very clear that the normal density consistently gives better results than the logistic density. The best transformation for MC is again “Normal + PCA”, but the best transformation for QMC is now “Normal + Cholesky”. We experimented with several values of λ for the logistic density and none of them give better results than the standard normal density. There must be some fundamental differences between the integrands of Examples 1 and 2 which are yet to be identified.

As in Fig. 6, the lattice rule “Lat1” appears to give worse results than other lattice rules, although the differences are not so noticeable here. With “Lat1” excluded, the other three lattice rules again appear to give slightly better results than the Sobol′ points. In terms of the reduction in standard errors, QMC beats MC (under its best transformation “Normal + PCA”) by a factor of 7.2 for “Normal + Cholesky”, 2.8 for “Normal + PCA”, 1.5 for “Logistic 0.6 + Cholesky”, and 1.4 for “Logistic 0.6 + PCA”. For this example, the MC standard errors can be more than seven times the QMC standard errors.

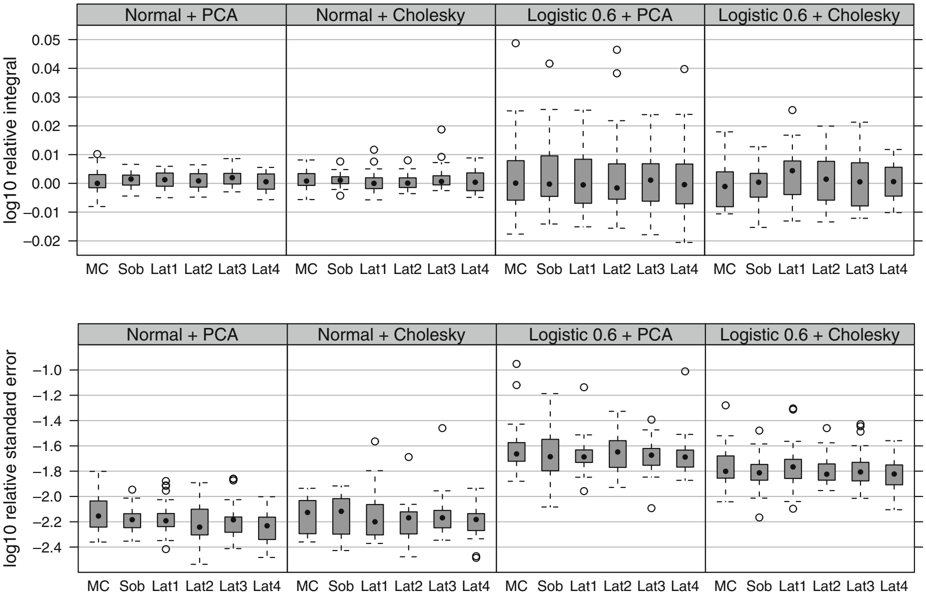

Fig. 8 Example 1 with $d = 200$. $\log_{10}$ relative integrals and $\log_{10}$ relative standard errors obtained from 30 repetitions based on ten random shifts of 65,536 points

Conclusion from Fig. 8

The results we have discussed so far correspond to a low nominal dimension of $d = 25$. Although $d = 25$ is reasonable for a random effect model such as Example 2, the dimension of interest for a time series model like Example 1 is often in the hundreds or thousands. Therefore we consider $d = 50, 100, 150, 200$ in Example 1 and carry out similar calculations as before (but with more integration points). It is noticeable that as the dimension d increases, the superiority of QMC over MC is dramatically reduced.

The results for $d = 200$ are summarised in Fig. 8. At $d = 200$, QMC performs only slightly better than MC under the same transformation, and the best transformation for both MC and QMC appears to be "Normal + PCA". The QMC results under "Logistic 0.6 + PCA" and "Logistic 0.6 + Cholesky" are in fact worse than the MC results under "Normal + PCA" or "Normal + Cholesky". The improvement of QMC over MC, both under "Normal + PCA", is a factor of merely 1.13.

To provide a readable scale, three outliers have been cropped out of the MC plot for "Normal + Cholesky": their $\log_{10}$ relative integral values are roughly 3.37, 2.91, and 1.88. These gross outliers correspond to integration points being extremely close to the boundaries of the unit cube where the transformed integrand is unbounded. Since the "Normal + Cholesky" transformation can lead to more problem boundaries (see the Appendix), it is understandable that we see more outliers in this case.

Recall that when $d = 25$ in Example 1 the best QMC transformation is "Logistic 0.6 + Cholesky". As d increases we observe that the best QMC transformation becomes "Normal + PCA". One possible explanation might be that in higher dimensions the region where the transformed integrand under "Normal + PCA" is unbounded is relatively tiny, while in most parts of the unit cube the integrand is almost constant. Thus the chance of getting an integration point in this tiny region is fairly slim. (We increased N to about one million and the best QMC transformation is still "Normal + PCA". However, with this many points, we ended up with many MC points close to the problem region under "Normal + Cholesky" and this resulted in massive outliers.)

6 Optimisation Using Quasi-Newton

The maximum likelihood approach seeks the parameter estimates $(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ that maximise the log-likelihood $\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma})$. Efficient numerical methods (see e.g. Fletcher 1987; Nocedal and Wright 1999) require at least the gradient and an approximation of the Hessian of $\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma})$. A *quasi-Newton method* uses difference in the gradients from successive iterations to build up an approximation to the Hessian, in combination with a line search to gain improvements in the objective. Here we provide only a brief outline of the optimisation process for Example 1 and highlight some important issues.

Let $\boldsymbol{\theta} = (\phi, \sigma, \beta)^\top$ denote the vector of parameters in Example 1, let $\ell(\boldsymbol{\theta})$ denote the log-likelihood associated with $\boldsymbol{\theta}$, and let $L(\boldsymbol{\theta}) = -\ell(\boldsymbol{\theta})/\ell(\boldsymbol{\theta}^{(0)})$ denote the objective function (which we aim to "minimise"), where $\boldsymbol{\theta}^{(0)}$ is our starting point, for example, the values based on the Laplace approximation. (Note that for both

Examples 1 and 2, the objective function values and its derivatives are expensive to obtain, but the number of parameters is relatively small.)

We need the gradient vector $\nabla L(\boldsymbol{\theta}) = (\frac{\partial L}{\partial \phi}, \frac{\partial L}{\partial \sigma}, \frac{\partial L}{\partial \beta})^\top$ as a function of $\boldsymbol{\theta}$. If $I \equiv I(\boldsymbol{\theta})$ denotes the integral $\int_{\mathbb{R}^d} \exp\{F(\mathbf{w})\}\, d\mathbf{w}$ as a function of $\boldsymbol{\theta}$, then we need to evaluate

$$\begin{aligned}\frac{\partial I}{\partial \phi} &= -\frac{\phi}{1-\phi^2}\, I - \frac{1}{2}\int_{\mathbb{R}^d} \exp\{F(\mathbf{w})\}\, \left(\mathbf{w}^\top \boldsymbol{\Sigma}_\phi^{-1}\mathbf{w}\right)\, d\mathbf{w},\\ \frac{\partial I}{\partial \sigma} &= -\frac{d}{\sigma}\, I - \frac{1}{2}\int_{\mathbb{R}^d} \exp\{F(\mathbf{w})\}\, \left(\mathbf{w}^\top \boldsymbol{\Sigma}_\sigma^{-1}\mathbf{w}\right)\, d\mathbf{w},\\ \frac{\partial I}{\partial \beta} &= (\mathbf{1}^\top \mathbf{y})\, I - e^\beta \int_{\mathbb{R}^d} \exp\{F(\mathbf{w})\}\, \left(\mathbf{1}^\top e^{\mathbf{w}}\right)\, d\mathbf{w},\end{aligned}$$

where

$$\boldsymbol{\Sigma}_\phi^{-1} := \frac{\partial \boldsymbol{\Sigma}^{-1}}{\partial \phi} = \frac{1}{\sigma^2}\begin{bmatrix} 0 & -1 & 0 & \cdots & 0 & 0\\ -1 & 2\phi & -1 & \cdots & 0 & 0\\ 0 & -1 & 2\phi & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & 2\phi & -1\\ 0 & 0 & 0 & \cdots & -1 & 0\end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_\sigma^{-1} := \frac{\partial \boldsymbol{\Sigma}^{-1}}{\partial \sigma} = -\frac{2}{\sigma}\boldsymbol{\Sigma}^{-1}.$$

Thus to obtain $L(\boldsymbol{\theta})$ and its gradient $\nabla L(\boldsymbol{\theta})$ for each $\boldsymbol{\theta}$, we need to evaluate the integral $I(\boldsymbol{\theta})$ plus three additional integrals.

Since the integrals will be estimated by the averages of transformed integrand values, to ensure that the estimate of the gradient vector is continuous and consistent, we need to use the same integration rule and the same transformation method for all four integrals. In other words, rather than finding the correct centre and scaling for each integrand separately, we use the centre and scaling obtained for the integrand in $I(\boldsymbol{\theta})$ in all four integral calculations. Fortunately, the integrands in these three additional integrals are just the integrand in $I(\boldsymbol{\theta})$ multiplied by simple terms like $\mathbf{w}^\top \boldsymbol{\Sigma}_\phi^{-1}\mathbf{w}$, $\mathbf{w}^\top \boldsymbol{\Sigma}_\sigma^{-1}\mathbf{w}$, and $\mathbf{1}^\top e^{\mathbf{w}}$, which do not have much impact on the feature of the integrand.

Starting with a parameter vector $\boldsymbol{\theta}^{(0)}$ and an approximate Hessian $\mathbf{B}^{(0)}$ at $\boldsymbol{\theta}^{(0)}$ (initially $\mathbf{B}^{(0)} = \mathbf{I}$ is often used), in step k we need to solve

$$\mathbf{B}^{(k)}\mathbf{d}^{(k)} = -\nabla L\left(\boldsymbol{\theta}^{(k)}\right)$$

for the search direction $\mathbf{d}^{(k)}$. Then the next iterate is given by

$$\boldsymbol{\theta}^{(k+1)} := \boldsymbol{\theta}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k)},$$

where $\alpha^{(k)}$ is the minimiser of $l_k(\alpha) = L(\boldsymbol{\theta}^{(k)} + \alpha\mathbf{d}^{(k)})$ obtained by an *approximate line search* based on modelling $l_k(\alpha)$ by a quadratic or cubic polynomial in α and then determining its stationary points by making use of $l'_k(\alpha) = {\mathbf{d}^{(k)}}^\top \nabla L(\boldsymbol{\theta}^{(k)})$. The approximate Hessian can be updated using, for example, the *BFGS update*

$$\mathbf{B}^{(k+1)} := \mathbf{B}^{(k)} + \frac{\mathbf{t}^{(k)}{\mathbf{t}^{(k)}}^\top}{{\mathbf{s}^{(k)}}^\top \mathbf{t}^{(k)}} - \frac{\mathbf{v}^{(k)}{\mathbf{v}^{(k)}}^\top}{{\mathbf{s}^{(k)}}^\top \mathbf{v}^{(k)}},$$

where

$$\begin{aligned}\mathbf{s}^{(k)} &:= \boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)} = \alpha^{(k)}\mathbf{d}^{(k)},\\ \mathbf{t}^{(k)} &:= \nabla L\left(\boldsymbol{\theta}^{(k+1)}\right) - \nabla L\left(\boldsymbol{\theta}^{(k)}\right),\\ \mathbf{v}^{(k)} &:= \mathbf{B}^{(k)}\mathbf{s}^{(k)} = -\alpha^{(k)}\nabla L\left(\boldsymbol{\theta}^{(k)}\right).\end{aligned}$$

The approximate line search conditions ensure that $\mathbf{s}^{(k)\top}\mathbf{t}^{(k)} > 0$ so that if $\mathbf{B}^{(0)}$ is positive definite then all subsequent $\mathbf{B}^{(k)}$ are positive definite. The quasi-Newton approximation satisfies the quasi-Newton relation $\mathbf{B}^{(k+1)}\mathbf{s}^{(k)} = \mathbf{t}^{(k)}$, thus $\mathbf{B}^{(k+1)}$ includes information from the difference in gradients $\mathbf{t}^{(k)}$ over the step $\mathbf{s}^{(k)}$.

We stress once again that it is important to use the same transformation and point set for both the objective value and its gradient, so that the derivatives are exact for the approximated objective. If this is not done, then both the line search and the gradient difference used to update the Hessian approximation can be affected. However, changing the parameter vector $\boldsymbol{\theta}^{(k)}$ changes the correct centre and scaling for the integrands. Experiments from the previous section show that the centring and rescaling play a crucial role in the transformation process. It is therefore essential that we carry out only a small number of iterations with a fixed centre and scaling to update the parameter vector. Then we restart the process with the correct centre and scaling to the updated parameter vector. The Hessian approximation obtained at the end of each round can be used as an initial approximation in the next round. Since the objective function arising from a log-likelihood model is often close to a convex quadratic at the optimal parameters, no more than $p+1$ steps of a p-parameter model should be done before updating the centre and scaling. Thus for the objective $L(\boldsymbol{\theta})$ in Example 1, we should carry out no more than 4 steps.

Another issue is when to stop the optimisation iteration. Convergence of an optimisation algorithm is typically based on the change in the objective function $L(\boldsymbol{\theta}^{(k+1)}) - L(\boldsymbol{\theta}^{(k)})$ and the norm $\|\nabla L(\boldsymbol{\theta}^{(k)})\|_\infty$ becoming small. However, it makes no sense trying to push these below the estimated standard error in the MC or QMC integral calculations. Thus as the optimisation iterates converge, a strategy for increasing the number of integration points is required.

Preliminary calculations indicate that a reasonable approximation of the maximum likelihood parameters can be obtained with just a few optimisation rounds, each with a small number of iteration steps. The Laplace approximation provides a very good starting point for the optimisation process. The best combination of optimisation steps with fixed centre and scaling and the strategy for increasing the number of integration points still requires further study.

7 Summary and Concluding Remarks

We carried out numerical experiments with the log-likelihood integrals from a Poisson state-space time series regression and a Poisson state-space linear mixed model. In both cases, the transformation which brings the integrand into the unit cube plays a crucial role because it determines the features of the transformed integrands. For small dimensions such as $d = 25$, QMC dramatically outperforms MC. However, as d increases to 200, the superiority of QMC over MC is diminished.

Values of d as small as 25 or 50 are realistic for the GLMMs and longitudinal data applications reviewed in Sections 2.1 and 2.2, and the results of this paper suggest QMC methods will lead to computational efficiencies in these settings.

For the pure time series setting reviewed in Section 2.3, we have demonstrated that the effective dimension of the integrand required to calculate the likelihood increases directly with the nominal dimension. This is in contrast with reported results for financial applications, in particular the valuation of an Asian option. In that setting the mean dimension of the integrand remains at around 2 for some forms of transformations and this may be due in part to the fact that the covariance structure of the multivariate normal is dominated by the first two or so principal components. That covariance structure corresponds to a non-stationary random walk. For the stationary autoregressive model considered here the covariance matrix is also increasingly dominated by a few principal components as ϕ increases to 1 and it might be speculated that the effective dimension will behave more like that observed for the financial application. For the mid-range value $\phi = 0.5$ used in this study the covariance matrix is not dominated by a few principal components and this may be a contributing factor to the effective dimension increasing directly with nominal dimension d. Additional research is required to better determine the impact of varying ϕ on the effective dimension and hence on the computational efficiency of QMC for the time series model.

We have not investigated the longitudinal data model discussed in Section 2.2. This model combines aspects of the models discussed in Section 2.3 (a pure time series model pursued as Example 1 in the paper) and Section 2.1 (a generalized linear mixed model pursued as Example 2 in the paper). Because of this the insights gained from studying Examples 1 and 2 are informative for the longitudinal model.

The two important statistical applications that we do consider in detail produce integrands that are considerably more challenging than the ones so far arising in finance. Future research in the application of QMC methodology to estimating integrals of the type that arise in highly structured generalised response models will require a more comprehensive understanding of the structure of financial type integrals and those considered here.

Acknowledgements The support of the Australian Research Council under its Centres of Excellence program is gratefully acknowledged. The first author was supported by a University of New South Wales Vice-Chancellor's Postdoctoral Research Fellowship. The authors are thankful for comments from Richard Davis, Yanan Fan, Berwin Turlach, Henryk Woźniakowski, Michael Giles, and the anonymous referee.

Appendix

A.1 Error Estimation Using Random Shifts

We now discuss how to use *randomly-shifted* QMC methods to provide error estimations.

Let $\boldsymbol{\Delta}$ denote a real vector in the unit cube, which we will refer to as the *shift*. The $\boldsymbol{\Delta}$-shift of a QMC method with points $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N$ is a QMC method with points

$$\mathrm{frac}(\boldsymbol{\xi}_i + \boldsymbol{\Delta}), \quad i = 1, \ldots, N,$$

where frac($\mathbf{x}$) is the vector obtained by replacing each component of $\mathbf{x}$ by its fractional part. It can be easily proved that the family of shifted QMC methods is an unbiased estimator of the true integral If.

In practice, for a chosen QMC method Q_N, we generate a number of independent random shifts $\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2, \ldots, \boldsymbol{\Delta}_r$ and form the approximations $Q_N^{(1)}, Q_N^{(2)}, \ldots, Q_N^{(r)}$, where $Q_N^{(k)}$ is the approximation of the integral If using a $\boldsymbol{\Delta}_k$-shift of Q_N. Then we take the average

$$\overline{Q}_N := \frac{1}{r}\left(Q_N^{(1)} + Q_N^{(2)} + \cdots + Q_N^{(r)}\right)$$

as our final approximation to If. An unbiased estimate for the standard error of $\overline{Q}_N$ is given by

$$\sqrt{\frac{1}{r}\frac{1}{r-1}\sum_{k=1}^{r}\left(Q_N^{(k)} - \overline{Q}_N\right)^2}.$$

A.2 Estimation of the Effective Dimensions

Here we outline the techniques described in Wang and Fang (2003) and Liu and Owen (2006) for estimating the effective dimensions. Using the identity

$$T_{\mathfrak{u}} := \sum_{v\subseteq \mathfrak{u}} \sigma^2(f_v) = \int_{[0,1]^{2d-|\mathfrak{u}|}} f(\mathbf{z})\, f(\mathbf{z}_{\mathfrak{u}}, \mathbf{w}_{-\mathfrak{u}})\, \mathrm{d}\mathbf{z}\, \mathrm{d}\mathbf{w}_{-\mathfrak{u}} - \left(\int_{[0,1]^d} f(\mathbf{z})\, \mathrm{d}\mathbf{z}\right)^2,$$

we can estimate the truncation dimension d_T by approximating $T_{\{1\}}$, $T_{\{1,2\}}$, $T_{\{1,2,3\}}$, …until the value reaches 99% of the total variance $\sigma^2(f)$. Here $(\mathbf{z}_{\mathfrak{u}}, \mathbf{w}_{-\mathfrak{u}})$ denotes the vector whose jth component is z_j if $j \in \mathfrak{u}$ and w_j if $j \notin \mathfrak{u}$. If $\{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N\}$ and $\{\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_N\}$ are the points of two d-dimensional MC or QMC rules, then $T_{\mathfrak{u}}$ can be approximated by

$$T_{\mathfrak{u}} \approx \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{\xi}_i)\, f(\boldsymbol{\xi}_{i,\mathfrak{u}}, \boldsymbol{\zeta}_{i,-\mathfrak{u}}) - (If)^2,$$

where If, as well as $\sigma^2(f)$, can be estimated using $\{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N\}$. Thus the cost of estimating d_T is the evaluation of at most $d+2$ integrals. The superposition dimension d_S can be estimated by making use of $T_{\mathfrak{u}}$ recursively. However, unless the integrals involved were computed very accurately, the propagation of the errors makes it impossible to get a realistic answer for d_S. The mean dimension d_M can be computed using the simple identity

$$d_M = \frac{1}{\sigma^2(f)}\sum_{\ell=1}^{d}\left(\frac{1}{2}\int_{[0,1]^{d+1}} \left(f(\mathbf{z}) - f(\mathbf{w}_{\{\ell\}}, \mathbf{z}_{-\{\ell\}})\right)^2 \mathrm{d}\mathbf{z}\, \mathrm{d}\mathbf{w}_{\{\ell\}}\right),$$

which requires the evaluation of $d+2$ integrals.

A.3 Component-by-Component Construction of Generating Vectors

Recall that the lattice points generated by $\boldsymbol{\eta}$ are given by

$$\boldsymbol{\xi}_i = \operatorname{frac}\left(\frac{i\boldsymbol{\eta}}{N}\right), \quad i = 1, \ldots, N.$$

For shifted rank-1 lattice rules with random shifts, we consider a "shift-averaged" worst-case error expression for our lattice rule generated by $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_d)$, which simplifies to (taking squares)

$$e_N^2(\eta_1, \ldots, \eta_d) = \frac{1}{N}\sum_{i=1}^{N} \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1,\ldots,d\}} \left(\gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} B_2\left(\operatorname{frac}\left(\frac{i\eta_j}{N}\right)\right)\right),$$

where $B_2(z) = z^2 - z + 1/6$ is the Bernoulli polynomial of degree 2. We shall not go into the detail of the derivation of this expression. For a given N and a set of weights $\gamma_{\mathfrak{u}}$, the components of a generating vector $\boldsymbol{\eta}$ will be constructed one at a time as follows:

1. Set $\eta_1 = 1$.
2. For each $d = 2, 3, \ldots$, with $\eta_1, \ldots, \eta_{d-1}$ fixed, choose η_d from the set $\{1 \leq \eta \leq N-1 : \gcd(\eta, N) = 1\}$ to minimise $e_N(\eta_1, \ldots, \eta_{d-1}, \eta_d)$.

This construction is now known in the QMC community as the *component-by-component construction* (e.g. Sloan et al. 2002). It has been proved that this construction leads to lattice rules that achieve an error of order $\mathcal{O}(N^{-1+\delta})$, $\delta > 0$, which is the optimal rate of convergence possible in our Sobolev space H (Kuo 2003). The implied constant in the big-$\mathcal{O}$ notation is independent of d provided that our weights $\gamma_{\mathfrak{u}}$ satisfy a certain condition.

Under the product weight setting, the total computational cost of this algorithm is only $\mathcal{O}(N\log(N)\,d)$ operations by making use of fast Fourier transforms (Nuyens and Cools 2006). Similar reductions in computational cost are also possible for the order-dependent weight setting.

A.4 Closer Examination of Transformation 1 with $b(x) = e^x$

Clearly the dominating terms in the exponent of f_1 are the exponential and the quadratic terms, which can be written together as

$$-\sum_{j=1}^{d} v_j \exp\left(\left(\mathbf{W}\mathbf{A}^*\Psi^{-1}(\mathbf{z})\right)_j\right) + \frac{1}{2}\sum_{j=1}^{d} v_j \left(\mathbf{W}\mathbf{A}^*\Psi^{-1}(\mathbf{z})\right)_j^2, \tag{12}$$

where $\mathbf{v} := \exp(\mathbf{X}\boldsymbol{\beta} + \mathbf{W}\mathbf{w}^*)$, and $(\mathbf{W}\mathbf{A}^*\Psi^{-1}(\mathbf{z}))_j$ denotes the jth component of $\mathbf{W}\mathbf{A}^*\Psi^{-1}(\mathbf{z})$.

Consider the situation where one component of $\mathbf{z}$, say z_k, goes to either 0 or 1, while all other components are away from the boundaries. Then the kth component of $\Psi^{-1}(\mathbf{z})$ goes to $\pm\infty$ and all other components are finite. In this case, the vector $\mathbf{WA}^*\Psi^{-1}(\mathbf{z})$ contains either 0 or $\pm\infty$ depending on the signs of the entries in the kth column of $\mathbf{WA}^*$. For example,

$$\underbrace{\begin{pmatrix} \cdot & \cdot & \cdot & 2 & \cdot \\ \cdot & \cdot & \cdot & -1 & \cdot \\ \cdot & \cdot & \cdot & 3 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 0 & \cdot \end{pmatrix}}_{\mathbf{WA}^*} \underbrace{\begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ +\infty \\ \cdot \end{pmatrix}}_{\Psi^{-1}(\mathbf{z})} = \underbrace{\begin{pmatrix} +\infty \\ -\infty \\ +\infty \\ +\infty \\ 0 \text{ or } +\infty \end{pmatrix}}_{\mathbf{WA}^*\Psi^{-1}(\mathbf{z})}$$

It is obvious that the expression (12) goes to $-\infty$ if any component of $\mathbf{WA}^*\Psi^{-1}(\mathbf{z})$ goes to $+\infty$; otherwise the expression goes to $+\infty$. Thus $f_1(\mathbf{z}) \to 0$ if any component of $\mathbf{WA}^*\Psi^{-1}(\mathbf{z})$ goes to $+\infty$; otherwise $f_1(\mathbf{z}) \to \infty$. To be more precise, the unboundedness occurs exactly when

- The kth column of $\mathbf{WA}^*$ contains no negative entry and $z_k \to 0$.
- The kth column of $\mathbf{WA}^*$ contains no positive entry and $z_k \to 1$.

Now we focus on Example 1 in which $\mathbf{W} = \mathbf{I}$. First we take $\mathbf{A}^*$ to be the PCA factor of $\mathbf{\Sigma}^*$. It turns out that only the first column of $\mathbf{A}^*$ contains all positive entries; all other columns contain entries with mixed signs. Thus $f_1(\mathbf{z})$ is only unbounded when $z_1 \to 0$. On the other hand, the Cholesky factor of $\mathbf{\Sigma}^*$ has no negative entry in any column. In this case $f_1(\mathbf{z})$ is unbounded whenever $z_k \to 0$ for any k.

A.5 Details About the Numerical Experiments in Section 5

Precisions, Random Number Generator, Inverse Normal, Sobol′ Points

The matrix and vector arithmetics are carried out using LAPACK (Anderson et al. 1999) in double precision. However, the accumulation of all integrals and the associated standard error estimates are done in long double precision.

We use a modified version of the random number generator ran2 by L'Ecuyer from Numerical Recipes in C (Press et al. 1995); we changed it to long double precision and we removed the guard for numbers getting close to 1. The inverse cumulative normal distribution function is computed using the online algorithm by Acklam (2007). It is accurate to full machine precision.

The parameters for the Sobol′ points are obtained from Joe and Kuo (2003), see also Bratley and Fox (1988).

Truncation and Mean Dimensions

Following Section A.2, we estimate the truncation and mean dimensions for Examples 1 and 2 using both Monte Carlo points and Sobol′ points. We use 500,000 points for $d = 25$, 50, 100 and 1,000,000 points for $d = 150$, 200. Each calculation is done with a single shift, and we repeat the calculation ten times. The number we report is the average of 20 results consisting of ten MC results and ten Sobol′ results.

Two-dimensional Projections

Figures 4 and 5 contain some two-dimensional projections corresponding to various transformed integrands for Example 1 with $d = 25$.

The graphs (a.1), (a.2), and (a.3) correspond to the transformed integrand obtained by using "Normal + PCA". In particular, graph (a.1) is the project of z_1 and z_5 through the centre of the unit cube $(0.5, \ldots, 0.5)$, graph (a.2) is the projection of z_3 and z_{25} through the point $(0.3, \ldots, 0.3)$, and graph (a.3) is the projection of z_7 and z_{11} through the point $(0.8, \ldots, 0.8)$. The graphs (b.1), (b.2), and (b.3) are obtained with the same parameters, except that they correspond to the transformed integrand obtained by using "Logistic 0.6 + Cholesky".

Graph (c) corresponds to the transformed integrand obtained without doing centring or rescaling. It uses the normal density and the PCA factor of the original covariance matrix $\mathbf{\Sigma}$. The graph is the projection of z_1 and z_5 through the centre of the unit cube.

Graph (d) corresponds to the transformed integrand obtained with centring but without rescaling. It uses the logistic density with $\lambda = 1$ and the PCA factor of the original covariance matrix $\mathbf{\Sigma}$. The graph is the projection of z_1 and z_5 through the centre of the unit cube.

Choices of Lattice Rules

In our numerical experiments we used lattice rules constructed from four different choices of weight settings, including two "finite-order and order-dependent weights", one "equal product weights", and one "decaying product weights", see Table 1.

Table 1 Lattice rules constructed based on four choices of weights

Lattice Rule	Weights	Description
Lat1	ORDER TWO $\Gamma_1 = \Gamma_2 = 1$, $\Gamma_\ell = 0\ \forall \ell \geq 3$	All second order interactions are equally important and there is no higher order interaction
Lat2	ORDER THREE 0.5 $\Gamma_1 = \Gamma_2 = 1, \Gamma_3 = 0.5$, $\Gamma_\ell = 0\ \forall \ell \geq 4$	The third order interactions are only half as important as the second order interactions and there is no higher order interaction
Lat3	EQUAL PRODUCT $\gamma_{\{j\}} = 0.5$, or $\Gamma_\ell = 0.5^\ell\ \forall \ell \geq 1$	All variables are equally important, and the higher order interactions become less and less important by a factor of 0.5
Lat4	DECAYING PRODUCT $\gamma_{\{j\}} = 1/j$	The importance of successive variables decays like $1/j$, and the weight associated with any group of variables is simply the product of the weights for those variables in this group

References

P. J. Acklam, “An algorithm for computing the inverse normal cumulative distribution function,” http://home.online.no/~pjacklam/notes/invnorm/, 2007.

E. Al-Eid, and J. Pan, “Estimation in generalized linear mixed models using SNTO approximation.” In A. R. Francis, K. M. Matawie, A. Oshlack, and G. K. Smyth (eds.), *Proceedings of the 20th International Workshop on Statistical Modelling,* pp. 77–84, Sydney, Australia, 2005.

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide, Third Edition*, SIAM: Philadelphia, 1999.

D. Bernat, W. T. M. Dunsmuir, and A. C. Wagenaar, “Effects of lowering the BAC limit to .08 on fatal traffic crashes in 19 states,” *Accident Analysis and Prevention* vol. 36 pp. 1089–1097, 2004.

J. G. Booth, J. P. Hobert, and W. Jank, “A survey of Monte Carlo algorithms for maximizing the likelihood of a two-stage hierarchical model,” *Statistical Modelling* vol. 1 pp. 333–349, 2001.

P. Bratley, and B. L. Fox, “Algorithm 659: Implementing Sobol's quasirandom sequence generator,” *ACM Transactions on Mathematical Software* vol. 14 pp. 88–100, 1988.

N. E. Breslow, and D. G. Clayton, “Approximate inference in generalized linear mixed models,” *Journal of the American Statistical Association* vol. 88 pp. 9–25, 1993.

BUGS Project, “BUGS: Bayesian Inference Using Gibbs Sampling,” http://www.mrc-bsu.cam.ac.uk/bugs, 2007.

R. E. Caflisch, W. Morokoff, and A. Owen, “Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension,” *Journal of Computational Finance* vol. 1 pp. 27–46, 1997.

B. S. Caffo, W. Jank, and G. L. Jones, “Ascent-based Monte Carlo expectation-maximization,” *Journal of the Royal Statistical Society Series B* vol. 67 pp. 235–251, 2005.

D. Clayton, “Generalized linear mixed models.” In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (eds.), *Markov Chain Monte Carlo in Practice*, pp. 275–301, Chapman & Hall: London, 1996.

R. Cools, F. Y. Kuo, and D. Nuyens, “Constructing embedded lattice rules for multivariate integration,” *SIAM Journal on Scientific Computing* vol. 28, pp. 2162–2188, 2006.

C. Crainiceanu, D. Ruppert, and M. P. Wand, “Bayesian analysis for penalised spline regression using WinBUGS,” *Journal of Statistical Software* vol. 14(14), 2005.

R. A. Davis, W. T. M. Dunsmuir, and Y. Wang, “Modelling time series of count data.” In S. Ghosh (ed.), *Asymptotics, Nonparametrics and Time Series*, pp. 63–114, Marcel-Dekker: New York, 1999.

R. A. Davis, W. T. M. Dunsmuir, and Y. Wang, “On autocorrelation in a Poisson regression model,” *Biometrika* vol. 87 pp. 491–505, 2000.

R. A. Davis, and G. Rodriguez-Yam, “Estimation for state-space models based on a likelihood approximation,” *Statistica Sinica* vol. 15 pp. 81–406, 2005.

J. Dick, F. Pillichshammer, and B. J. Waterhouse, “The construction of good extensible rank-1 lattices,” *Mathematics of Computation*, (in press), 2007.

P. Diggle, K.-L. Liang, and S. Zeger, *Analysis of Longitudinal Data*, Oxford University Press: Oxford, 1995.

P. Diggle, P. Heagerty, K.-L. Liang, and S. Zeger, *Analysis of Longitudinal Data, Second Edition*, Oxford University Press: Oxford, 2002.

M. Evans, and T. Swartz, “Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems,” *Statistical Science* vol. 10 pp. 254–272, 1995.

R. Fletcher, *Practical Methods of Optimisation, Second Edition*, Wiley: Chichester, 1987.

J. González, F. Tuerlinckx, P. De Boeck, and R. Cools, “Numerical integration in logistic-normal models,” *Computational Statistics and Data Analysis* vol. 51 pp. 1535–1548, 2006.

L. C. Gurrin, K. J. Scurrah, and M. L. Hazelton, “Tutorial in biostatistics: Spline smoothing with linear mixed models,” *Statistics in Medicine* vol. 24 pp. 3361–3381, 2005.

F. J. Hickernell, and H. S. Hong, “Quasi-Monte Carlo methods and their randomisations.” In R. Chan, Y.-K. Kwok, D. Yao, and Q. Zhang (eds.), *Applied Probability*, AMS/IP Studies in Advanced Mathematics, vol. 26, pp. 59–77, American Mathematical Society: Providence, 2002.

F. J. Hickernell, C. Lemieux, and A. B. Owen, “Control variates for quasi-monte carlo,” *Statistical Science* vol. 20 pp. 1–31, 2005.

W. Jank, “Quasi-Monte Carlo sampling to improve the efficiency of Monte Carlo EM,” *Computational Statistics and Data Analysis* vol. 48 pp. 685–701, 2005.

S. Joe, and F. Y. Kuo, “Remark on Algorithm 659: Implementing Sobol’s quasirandom sequence generator,” *ACM Transactions on Mathematical Software* vol. 29 pp. 49–57, 2003.

A. Y. C. Kuk, “Laplace importance sampling for generalized linear mixed models,” *Journal of Statistical Computation and Simulation* vol. 63 pp. 143–158, 1999.

F. Y. Kuo, “Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces,” *Journal of Complexity* vol. 19 pp. 301–320, 2003.

F. Y. Kuo, G. W. Wasilkowski, and B. J. Waterhouse, “Randomly-shifted lattice rules for unbounded integrands,” *Journal of Complexity* vol. 22 pp. 630–651, 2006.

F. Y. Kuo, and I. H. Sloan, “Lifting the curse of dimensionality,” *Notices of the American Mathematical Society* vol. 52 pp. 1320–1328, 2005.

P. L’Ecuyer, and C. Lemieux, “Recent advances in randomized quasi-Monte Carlo methods.” In M. Dror, P. L’Ecuyer, and F. Szidarovszki (eds.), *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*, pp. 419–474, Kluwer Dordrecht, 2002.

R. Liu, and A. Owen, “Estimating mean dimensionality of analysis of variance decompositions,” *Journal of the American Statistical Association* vol. 101 pp. 712–721, 2006.

C. E. McCulloch, and S. R. Searle, *Generalized, Linear, and Mixed Models*, Wiley: New York, 2000.

H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM: Philadelphia, 1992.

H. Niederreiter, “Construction of (t, m, s)-nets and (t, s)-sequences,” *Finite Fields and Their Applications* vol. 11, pp. 578–600, 2005.

J. Nocedal, and S. J. Wright, *Numerical Optimization*, Springer, 1999.

D. Nuyens, and R. Cools, “Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces,” *Mathematics of Computation* vol. 75 pp. 903–920, 2006.

A. R. Owen, and S. D. Tribble, “A quasi-Monte Carlo Metropolis algorithm,” *Proceedings of the National Academy of Sciences* vol. 102 pp. 8844–8849, 2005.

J.-X. Pan, and R. Thompson, “Quasi-Monte Carlo EM algorithm for estimation in generalized linear mixed models.” In R. Payne, and P. Green (eds.), *Proceedings in Computational Statistics*, pp. 419–424, Physical-Verlag, 1998.

J. Pan, and R. Thompson, “Quasi-Monte Carlo estimation in generalized linear mixed models.” In A. Biggeri, E. Dreassi, C. Lagazio, M. Marchi (eds.), *Proceedings of the 19th International Workshop on Statistical Modelling,* pp. 239–243, Firenze University Press: FLorence, 2004.

J. Pan, and R. Thompson, “Quasi-Monte Carlo approximation for estimation in generalized linear mixed models,” *Computational Statistics and Data Analysis* vol. 51 pp. 5765–5775, 2007.

S. H. Paskov, and J. F. Traub, “Faster valuation of financial derivatives,” *Journal of Portfolio Management* vol. 22 pp. 113–120, 1995.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C, Second Edition*, Cambridge University Press, 1995.

D. Ruppert, M. P. Wand, and R. J. Carroll, *Semiparametric Regression*, Cambridge University Press: New York, 2003.

SAS Institute, Inc, http://www.sas.com, 2007.

A. Skrondal, and S. Rabe-Hesketh, *Generalized Latent Variable Modelling: Multilevel, Longitudinal and Structural Equation Models*, Chapman and Hall: Boca Raton, Florida, 2004.

I. H. Sloan, and S. Joe, *Lattice Methods for Multiple Integration*, Oxford University Press: Oxford, 1994.

I. H. Sloan, F. Y. Kuo, and S. Joe, “Constructing randomly shifted lattice rules in weighted Sobolev spaces,” *SIAM Journal on Numerical Analysis* vol. 40 pp. 1650–1665, 2002.

I. H. Sloan, X. Wang, and H. Woźniakowski, “Finite-order weights imply tractability of multivariate integration,” *Journal of Complexity* vol. 20 pp. 46–74, 2004.

I. H. Sloan, and H. Woźniakowski, “When are quasi-Monte Carlo algorithms efficient for high dimensional integrals?” *Journal of Complexity* vol. 14 pp. 1–33, 1998.

M. P. Wand, “Smoothing and mixed models,” *Computational Statistics* vol. 18 pp. 223–249, 2003.

X. Wang, and K. T. Fang, “Effective dimensions and quasi-Monte Carlo integration,” *Journal of Complexity* vol. 19 pp. 101–124, 2003.

R. Wolfinger, and M. O’Connell, “Generalized linear mixed models: a pseudo-likelihood approach,” *Journal of Statistical Computation and Simulation* vol. 48 pp. 233–243, 1993.

Y. Zhao, J. Staudenmayer, B. A. Coull, and M. P. Wand, “General design Bayesian generalized linear mixed models,” *Statistical Science* vol. 21 pp. 35–51, 2006.