Research Article

Statistics in Medicine

Received 14 July 2010, Accepted 7 March 2011 Published online 18 May 2011 in Wiley Online Library

(wileyonlinelibrary.com) DOI: 10.1002/sim.4254

Fitting dynamic models with forcing functions: Application to continuous glucose monitoring in insulin therapy

D. J. Lunn,[a*†] C. Wei[a] and R. Hovorka[b]

The artificial pancreas is an emerging technology to treat type 1 diabetes (T1D). It has the potential to revolutionize diabetes care and improve quality of life. The system requires extensive testing, however, to ensure that it is both effective and safe. Clinical studies are resource demanding and so a principle aim is to develop an *in silico* population of subjects with T1D on which to conduct pre-clinical testing. This paper aims to reliably characterize the relationship between blood glucose and glucose measured by subcutaneous sensor as a major step towards this goal. Blood- and sensor-glucose are related through a dynamic model, specified in terms of differential equations. Such models can present special challenges for statistical inference, however. In this paper we make use of the BUGS software, which can accommodate a limited class of dynamic models, and it is in this context that we discuss such challenges. For example, we show how dynamic models involving forcing functions can be accommodated. To account for fluctuations away from the dynamic model that are apparent in the observed data, we assume an autoregressive structure for the residual error model. This leads to some identifiability issues but gives very good predictions of virtual data. Our approach is pragmatic and we propose a method to mitigate the consequences of such identifiability issues. Copyright © 2011 John Wiley & Sons, Ltd.

Keywords: dynamic models; glucose kinetics; artificial pancreas; autoregressive models; WinBUGS

1. Introduction

Type 1 diabetes (T1D) is a chronic autoimmune disorder characterized by dysregulated blood-glucose (BG) levels due to an inability of the pancreas to produce insulin, the hormone that promotes uptake of glucose by cells [1]. Persistent exposure to high glucose levels (hyperglycaemia) causes long-term diabetes complications and organ dysfunction [2]. The standard therapy is based on multiple insulin injections, using a combination of short and long acting insulin analogues, informed by frequent BG self-monitoring [3]. Treatment by continuous subcutaneous insulin infusion (CSII [4]) is on the rise and uses a portable electromechanical pump to mimic nondiabetic insulin delivery, infusing insulin at preselected rates—basically a slow basal rate with patient-activated boosts at mealtimes. However, intensive insulin therapy aiming to achieve near-normal glucose control is associated with an increased risk of low BG levels (hypoglycaemia), potentially leading to seizures, unconsciousness, brain damage and even death [5]. Optimization of insulin therapy is confounded by large day-to-day and diurnal variability in insulin requirements influenced by factors such as exercise, stress, and recurrent illness [6–8].

Self-monitoring of BG offers only a snapshot, each time, of the underlying glucose excursion, thus, making for considerable uncertainty in determining the right treatment decision to achieve and

[a] *Medical Research Council Biostatistics Unit, Institute of Public Health, University Forvie Site, Robinson Way, Cambridge CB2 0SR, U.K.*

[b] *Diabetes Modelling Group, Department of Paediatrics, University of Cambridge, Cambridge, U.K.*

[*] *Correspondence to: D. J. Lunn, Medical Research Council Biostatistics Unit, Institute of Public Health, University Forvie Site, Robinson Way, Cambridge CB2 0SR, U.K.*

[†] *E-mail: david.lunn@mrc-bsu.cam.ac.uk*

maintain desirable glucose levels. Continuous glucose monitoring (CGM [9]) devices offer an alternative approach informing on real-time glucose levels, with the possibility of real-time hyperglycaemia and hypoglycaemia alerts [10]. CGM can show the rate at which glucose is increasing or decreasing, and thus facilitate an understanding of how glucose levels react to insulin, food, exercise, and other factors, thus, providing scope for finer glucose control. CGM devices use a sensor to measure *interstitial* glucose, which provides an indirect reading of BG. That they can lead to improved glycaemic control has been demonstrated [11]. They may also reduce the frequency of hypoglycaemia events in people with well controlled T1D [12]. CGM devices and insulin pumps can be combined to form an artificial pancreas. Insulin delivery is then automatically modulated according to real-time sensor-glucose (SG), as directed by a control algorithm, rather than at preselected rates as during the conventional 'open-loop' CSII [13].

An impediment to real-time accurate CGM tracing is the existence of a physiological delay between BG and interstitial glucose [14–16]. While there is great potential for CGM systems and the artificial pancreas to revolutionize diabetes care and improve quality of life, a more detailed quantitative understanding is needed about the relationship between BG and SG as reported by CGM devices. This information is helpful to health-care professionals, subjects with T1D and their carers, and also to facilitate *in silico* testing of the artificial pancreas [17]. With this latter objective in mind, it is important to be able to predict *realistic* sensor data, since sensor data are what the control algorithm will have to respond to. Our approach is to characterize the relationship between BG and SG throughout the population using data from a relatively small clinical study. We use a nonlinear regression function to describe each individual's data and simultaneously estimate the population distribution of the underlying parameters, exploring both inter- and intra-individual variability, including any correlations between the parameters. We also use an autoregressive (AR) process to accurately describe the residual errors. The estimated population distribution can then be used to simulate realistic parameter sets for new individuals, with appropriate correlations between parameters. These can then be used, in combination with real or simulated BG data, to predict SG profiles for the new individuals, which are overlaid with simulated AR processes (informed by the estimated model) to account for typical differences between sensor observations and our regression function. A wealth of virtual data can thus be generated, allowing extensive testing and accelerated development of the artificial pancreas [17].

Existing CGM devices lack the accuracy of BG meters. Early reports documented particular concerns at low BG values [18, 19], although more recent assessments indicate comparable relative accuracy at normal and low glucose ranges [20, 21]. Each new generation of CGM devices brings about improvements in accuracy, reliability, and sensitivity and specificity of hypo- and hyperglycaemia alerts [10, 21, 22]. This has a positive effect on the utility and frequency of CGM use, which is associated with health benefits [23] and facilitates the development of the artificial pancreas [24, 25]. Our work complements these developments by providing a methodological framework and insights into the nature and statistical properties of sensor errors. This may also inform the development of advanced control algorithms for the artificial pancreas [26].

The present paper is concerned with the Guardian® RT CGM system [27]. Breton and Kovatchev [28] use similar ideas to model another CGM system (FreeStyle Navigator™, Abbott Diabetes Care, Alameda, California). However, their estimation strategy is somewhat fragmented, with parameter uncertainty being ignored between the various stages. We extend their approach by combining all of the various modelling components into a single model, allowing all sources of uncertainty to propagate through to our final inferences. We also examine the process of calibrating the sensor in more detail, allowing the model to be extended to handle multiple calibration events for a single individual. Finally, we simultaneously model the inter- and intra-individual variability of system parameters in order to facilitate prediction.

The structure of the paper is as follows. In Section 2 we describe the data obtained from a small clinical study involving 12 children and adolescents. Section 3 provides some mathematical background and then presents the various aspects of our statistical model, including the dynamic sub-model and how we account for system calibration. Section 4 presents the results of our analyses, and a concluding discussion is given in Section 5.

2. Data

A total of $N = 12$ children and adolescents with T1D treated by continuous subcutaneous insulin infusion participated in a clinical research study conducted at the Wellcome Trust Clinical Research

Facility, Addenbrooke's Hospital, University of Cambridge, UK [24]. The sample size was not based on any power calculation as the study concerned (APCam01 in [24]) was exploratory. Participants and, as appropriate, their carers gave informed consent/assent. The study was approved by the Cambridge Research Ethics Committee (REC Ref 06/Q0108/350).

A glucose sensor was fitted to each participant at least 24 h prior to the study, and, following a run-in period and calibration as suggested by the manufacturer, the Guardian® RT CGM system took SG readings every 5 min. BG was measured every 15 min by collecting samples via a venous cannula. The study ran from 17:00 until 12:00 the following day, giving $m=77$ BG and $n=228$ SG measurements for each individual. There are a small number of missing SG measurements, which we shall treat as unknown parameters in order to retain a balanced data set. Two self-selected meals were eaten at 18:00 and 08:00 the following morning to maintain a normal carbohydrate intake, each meal containing a mean (SD) of 87 (23) g carbohydrates. Prandial insulin boluses were given with the meals. During the study, the Guardian® RT was calibrated, using recent BG measurements, shortly after 17:00 and every 6 h thereafter, thus splitting the study period into five distinct 'calibration periods' for each individual.

3. Methods

3.1. Background

The model that we will describe in subsequent sections is essentially an extension of the non-linear regression

$$y_j = g(\theta, x_j) + \varepsilon_j, \quad j = 1, \ldots, n,$$

where y_j and x_j denote response and independent variables, respectively, θ denotes a set of regression parameters, and $\varepsilon_j \sim \mathrm{N}(0, \sigma^2)$, say. As a simple example, let us consider a situation in which the regression function $g()$ represents exponential decay and x is elapsed time:

$$g(\theta, x) = \theta_1 \exp(-\theta_2 x). \tag{1}$$

Note that (1) is the *unique* solution to

$$\frac{\mathrm{d}g}{\mathrm{d}x} = -\theta_2 g, \quad g(\theta, x=0) = \theta_1, \tag{2}$$

and so (1) and (2) are equivalent specifications of the same regression function. Now suppose that we can only express our regression function in terms of differential equations (with the corresponding initial conditions) as we do not know the analytic solution. If we know that a unique solution exists, however, then we know that solution is simply a deterministic function of the inputs, θ and x, albeit of unknown form. If we can find a way to evaluate the solution, we may thus treat it as we would any other deterministic function. We may then exploit standard graphical modelling theory [29, 30] to evaluate the full conditional distributions of any unknown inputs, e.g. θ_2. In this paper, we make use of the BUGS software [31, 32] with *WBDiff* interface [33] installed (to allow specification of differential equations). The differential equations, described in the following subsection, are solved numerically by the software using a Runge–Kutta algorithm [34], and Metropolis–Hastings samplers [35, 36] are typically used for sampling the unknown inputs.

Now suppose that the differential equations depend on some additional quantity, such as the ambient temperature or pressure, say, whose evolution through time is driven by external factors. This happens in many settings. For example, wind stress may be a factor in modelling ocean circulation [37], whereas light intensity, temperature, availability of food/nutrients, and wind speed may all be important in ecological modelling [38]. In our case, the equations depend on BG concentrations but in other areas of diabetes research insulin concentrations may be used, e.g. [39]. We cannot usually model such quantities but may be able to observe their values over a series of times. If we interpolate between the observations, then we can approximate the relevant quantity at any time within the observation period. If this observation period envelopes the time-frame over which we wish to evaluate our regression function, then solving the differential equations is still possible, and the interpolated series is referred to as a *forcing function*. WBDiff has not been designed with the specification of forcing functions in mind, but we show how they may be accommodated in Section 3.5.

As we shall see later, our regression function (also referred to herein as the 'dynamic model') is somewhat imperfect—there are clear fluctuations away from the fitted model apparent in the observed data. For the purposes of testing the artificial pancreas' control algorithm, it is important that we are able to predict realistic sensor data, and so we consider an AR model for the residual errors (ε_j above)—see Section 3.3. We fit the AR and dynamic models simultaneously, to fully account for uncertainty and also to capture posterior correlation between them. However, we find that they are somewhat confounded and identifiability issues arise unless the degree of autocorrelation in the AR process is constrained. Our approach to this is pragmatic and involves exploring various ways of limiting the extent to which the two models may interact, as discussed at the end of Section 4 and in the discussion.

3.2. *Glucose kinetics*

The CGM sensor measures glucose concentrations in the interstitial fluid, which can be related, mathematically, to the BG concentration via a compartmental model [40, 41]:

$$\frac{\mathrm{dIG}(t)}{\mathrm{d}t} = -p_1 \mathrm{IG}(t) + p_2 \mathrm{BG}(t), \tag{3}$$

where IG denotes interstitial glucose. Hence, IG increases at a rate proportional to BG but is 'used up' according to a first-order process—the more there is, the faster it disappears.

The sensor does not measure IG directly but, instead, measures electric current in the interstitial fluid and maps this to a *scaled* measure of IG via an assumption of proportionality. When the system is calibrated, the appropriate scale is chosen by equating scaled current with recent measures of BG. To account for this calibration we transform (3) to the same scale, by defining *normalized* interstitial glucose NIG$=v$IG and choosing $v = p_1/p_2$ so that NIG is equal to BG at steady state:

$$\frac{\mathrm{dNIG}(t)}{\mathrm{d}t} = -p_1\{\mathrm{NIG}(t) - \mathrm{BG}(t)\}. \tag{4}$$

Let SG_{ij} denote the jth measured SG concentration for individual $i (i = 1, \ldots, N = 12, j = 1, \ldots, n = 228)$. Similarly, let $\mathrm{BG}_{il}, l = 1, \ldots, m = 77$, denote the lth measured BG concentration for individual i. Further, denote the times at which SG_{ij} and BG_{il} were measured by t_{ij} and s_{il}, respectively. A simple model for fitting individual i's data is then

$$\mathrm{SG}_{ij} = \mathrm{NIG}_{ij} + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathrm{N}(0, \sigma_i^2), \tag{5}$$

where NIG_{ij} is the solution to (4) at time t_{ij}. This is a deterministic function of three unknown inputs: (i) the value of p_1; (ii) the initial condition $\mathrm{NIG}(t=0)$; and (iii) the form of $\mathrm{BG}(t)$. We assume that each individual has a distinct, but unknown, value of p_1, which we denote by p_{1_i}. Often the initial conditions will be known, but in general they are not, and so these may also be treated as unknown parameters; in this case denoted by $\mathrm{NIG}_{0_i}, i = 1, \ldots, N$. Regarding the form of $\mathrm{BG}(t)$, we assume that linearly interpolating[‡] between the observed values for each individual, $\mathrm{BG}_{il}, l = 1, \ldots, m$, provides a satisfactory approximation, although see later for further discussion. Denoting the forcing function for individual i by $\mathrm{BG}_i(t)$, we then have

$$\mathrm{BG}_i(t) = \mathrm{BG}_{ip} + \frac{\mathrm{BG}_{i(p+1)} - \mathrm{BG}_{ip}}{s_{i(p+1)} - s_{ip}} \times (t - s_{ip}), \quad p = \sum_{l=1}^{m-1} l \times I(s_{il} \leqslant t < s_{i(l+1)}).$$

3.3. *Calibration*

As demonstrated in Figure 1(b), the simple model above can perform poorly. With some careful thought, however, as to the nature of the underlying calibration mechanism, we can do much better. We stress, though, that the details of calibration are actually unknown to us—implementation details of the calibration procedure are proprietary and comprise guarded know-how by the respective CGM-system manufacturers to retain competitive advantage. In what follows, we make basic assumptions about how the process *might* work in order to construct a reasonable model. We first assume that SG is given by $A \times I_m + D$, where A and D are unknown constants and I_m is the electrical current measured by the

[‡] The numerical differential equation solver needs to be able to evaluate $\mathrm{BG}(t)$ at any time, not just those times at which BG has been observed.

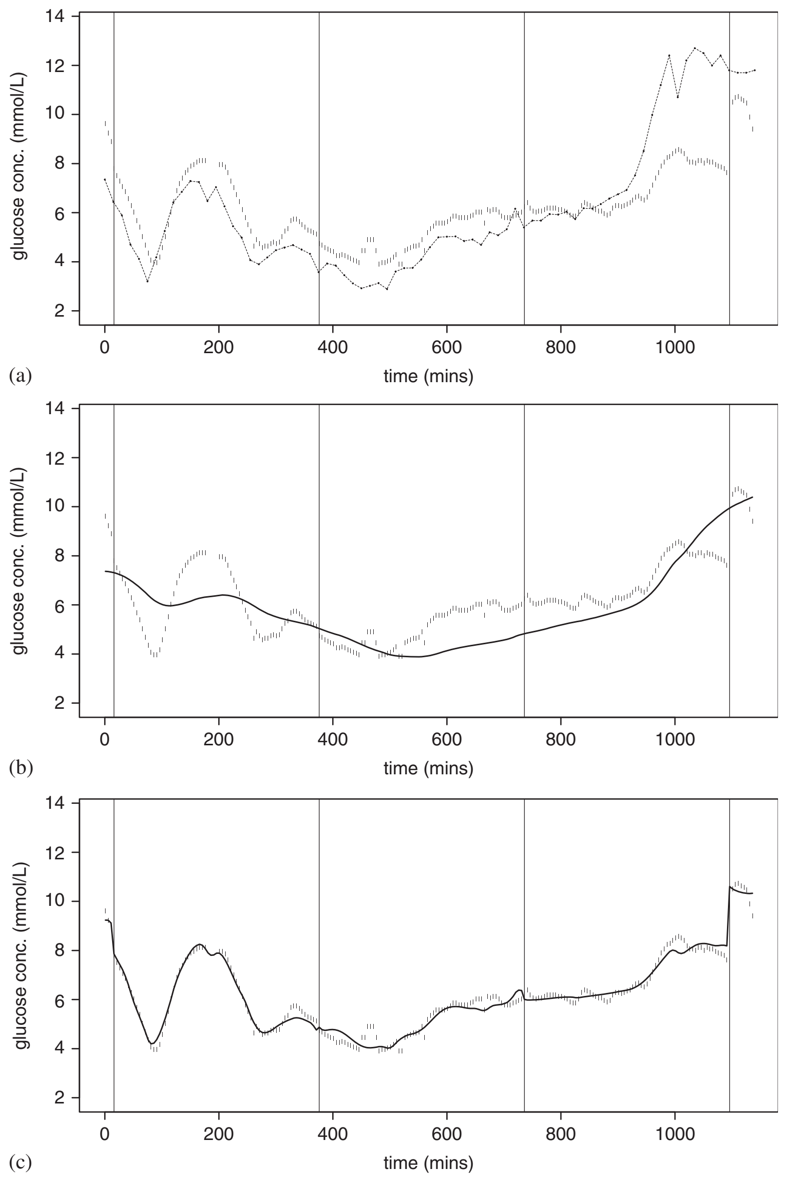

Figure 1. Observed data and posterior median model-predicted concentrations for individual ‘10’ plotted against time since beginning of study: (a) observed BG (—·—) and SG (|) concentrations; (b) observed SG concentrations (|) and model-predicted values NIG_{10j} (—) from ‘basic’ population model; and (c) observed SG concentrations (|) and model-predicted values CIG_{10j} (—) from ‘calibrated’ population model.

sensor, which we assume is subject to some error, δ, such that $I_m = I + \delta$, where I is the true current. We also assume that the true current I is related to interstitial glucose IG through $I = \text{IG}/S + I_B$, where S denotes *current sensitivity* and I_B represents a baseline current that is present even in the absence of IG. Hence,

$$\begin{aligned}\text{SG} &= \frac{A}{Sv}\text{NIG} + A.I_B + A.\delta + D\\ &= F\cdot\text{NIG} + B + \eta = \text{CIG} + \eta,\end{aligned}$$

where $\text{CIG} = F\cdot\text{NIG} + B$ denotes ‘calibrated interstitial glucose’. Note that A, S, v, I_B, δ, and D are all unknown and so only $F = A/Sv$, $B = A\cdot I_B + D$, and $\eta = A\cdot\delta$ can feasibly be identified. Note also that while S, v, I_B, and $\text{Var}(\delta)$ might all reasonably be assumed constant, both A and D change every time the sensor is calibrated, leading to new values for F, B, and $\text{Var}(\eta)$ anyway.

Let ψ_{ik} denote the kth calibration time for individual $i (k=1,\dots,K=4)$. For convenience, we also define $\psi_{i0}=t_{i1}$ and $\psi_{i(K+1)}>t_{in}$. We can then write down the calibration period to which each SG_{ij} belongs as

$$P[i,j]=\sum_{k=1}^{K+1} k\times I(\psi_{i(k-1)}\leqslant t_{ij}<\psi_{ik}).$$

(In this paper, the calibration times are assumed known, but, in general, we may wish to acknowledge some uncertainty regarding their values.) A more realistic model for SG observations is then

$$\mathrm{SG}_{ij}=F_{i\,P[i,j]}\times \mathrm{NIG}_{ij}+B_{i\,P[i,j]}+\eta_{ij}=\mathrm{CIG}_{ij}+\eta_{ij}, \tag{6}$$

where B_{ik} and $F_{ik}(k=1,\dots,K+1)$ are unknown, individual, and calibration-period-specific parameters referred to as the calibration shift and calibration scale-factor, respectively.

We consider two different models for the residuals η_{ij}. Assuming that the only source of 'error' is the measured current, then a homoscedasticity assumption leads to $\eta_{ij}\sim \mathrm{N}(0,\sigma^2_{i\,P[i,j]})$, $i=1,\dots,N$, $j=1,\dots,n$, whereas an AR model gives

$$\eta_{ij}=\rho_i\frac{F_{i\,P[i,j]}}{F_{i\,P[i,j-1]}}\eta_{i(j-1)}+\gamma_{ij},\quad i=1,\dots,N,\ j=2,\dots,n, \tag{7}$$

with $\gamma_{ij}\sim \mathrm{N}(0,\sigma^2_{i\,P[i,j]})$, $i=1,\dots,N$, $j=1,\dots,n$. Here ρ_i is an unknown individual-specific parameter controlling the degree of autocorrelation among the residuals. Note that in order to obtain the $F_{i\,P[i,j]}/F_{i\,P[i,j-1]}$ term, which is equal to one except at the calibration times, when it 'adjusts' the AR process, we make the assumption that current sensitivity, S, and v do not change over time for a given individual. We cannot specify the autoregressive model, as given by (6)–(7), in BUGS, however, since a logical relationship for the response variable is not allowed. One way to get around this is to assume that each individual's SG-series arises from a CAR distribution [42, 43]. A more flexible and intuitive approach, though, is to reexpress (6)–(7) as

$$\mathrm{SG}_{ij}\sim \mathrm{N}(\phi_{ij},\sigma^2_{i\,P[i,j]}),\quad \phi_{ij}=\mathrm{CIG}_{ij}+\rho_i\frac{F_{i\,P[i,j]}}{F_{i\,P[i,j-1]}}\eta_{i(j-1)},\quad \eta_{ij}=\mathrm{SG}_{ij}-\mathrm{CIG}_{ij},$$

for $j=2,\dots,n$.

To define η_{i1}, we could simply specify $\eta_{i1}=\mathrm{SG}_{i1}-\mathrm{CIG}_{i1}$. However, note that the AR model does not penalize large ηs, and so unless we control their size, through an informative prior on η_{i1}, say, they can become large and force the underlying 'model fit' $\{\mathrm{CIG}_{ij}, j=1,\dots,n\}$ away from the observed data, leading to implausible parameter estimates (see later for discussion). Note, though, that as the initial condition for the differential equation is unknown, we are free to choose the time to which it relates. If we choose the time of the first sensor reading t_{i1}, then we may express the initial condition deterministically:

$$\mathrm{NIG}_{0_i}=\mathrm{NIG}_{i1}=(\mathrm{SG}_{i1}-\eta_{i1}-B_{i1})/F_{i1}.$$

Hence, specifying a prior for η_{i1} means that there is no need to model the initial conditions. Figure 2 shows a graphical representation of the full model in the case of autoregressive errors.

3.4. Priors

Calibration shifts and scale-factors may be correlated; in addition, scale factors must be positive whereas shifts may be negative. We therefore define $C_{ik}=\binom{\log F_{ik}}{B_{ik}}$, $i=1,\dots,N, k=1,\dots,K+1$, which we assume arise from a bivariate normal 'population' distribution. If we believe that calibration parameters reflect characteristics of the individual (and/or sensor), then we may wish to assume individual-specific means ω_i and an intra-individual covariance Ω: $C_{ik}\sim \mathrm{MVN}_2(\omega_i,\Omega)$. We may then wish to assume that the ω_is also arise from a bivariate normal distribution, with unknown 'global' mean μ and *inter*-individual covariance Σ:

$$\omega_i\sim \mathrm{MVN}_2(\mu,\Sigma). \tag{8}$$

If, on the other hand, we believe that there is no correlation among calibration vectors for the same individual, then we might assume that it is the C_{ik}s that are drawn from the right-hand side of (8) instead.

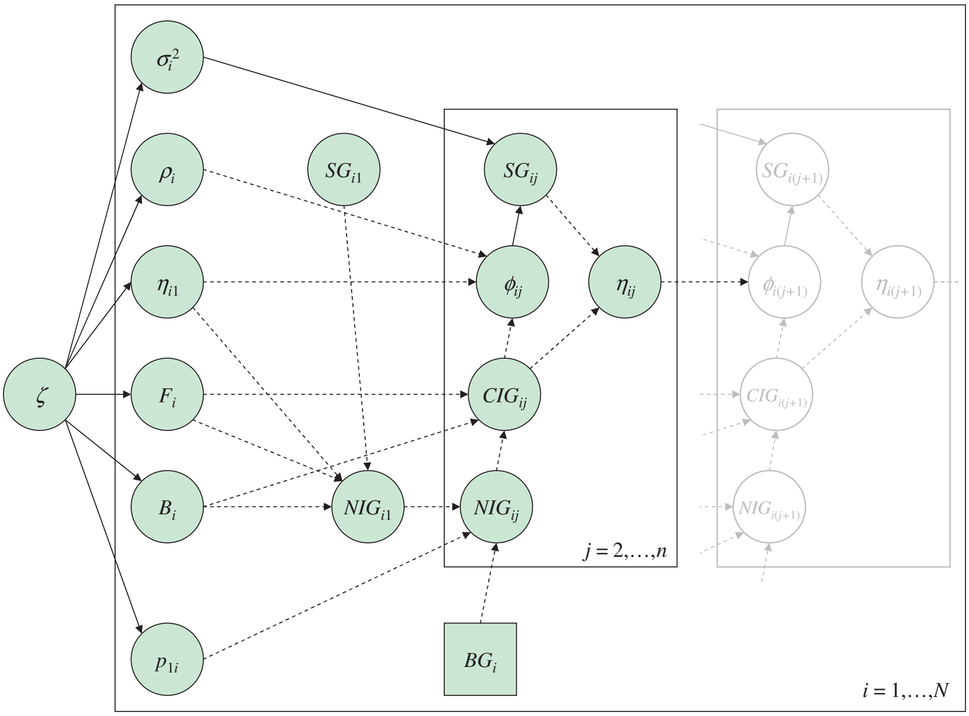

Figure 2. Directed acyclic graph (DAG) corresponding to 'calibrated' model with AR(1) process for the residual errors. For simplicity, the case in which there is only one calibration period for each individual is depicted. Each variable in the statistical model corresponds to a *node* and links between nodes show direct dependence. The graph is *directed* because each link is an arrow; it is *acyclic* because by following the arrows it is not possible to return to a node after leaving it. Square nodes denote known constants whereas circular nodes represent either deterministic relationships (i.e. functions) or stochastic quantities, i.e. quantities that require a distributional assumption. Stochastic dependence and functional dependence are denoted by solid and dashed arrows, respectively. Repetitive structures, such as the 'loop' from $i=1$ to $i=N$, are represented by 'plates', which are nested if the model is hierarchical. The 'plate' in light-type on the right-hand side is shown to indicate the nature of dependence between successive observations. Nodes ζ and BG_i denote the entire set of population parameters, and the set of observed blood glucose concentrations for individual i, $\{\text{BG}_{il}, l=1,\ldots,m\}$, respectively.

We assume fairly standard, vague (but proper) priors for μ, Ω, and Σ: bivariate normal and inverse-Wishart, centered at our best *a priori* guess with large variance. Throughout, the prior standard deviation specified for vague normal priors is 100 whereas Wishart priors are made as vague as possible by setting the degrees of freedom equal to the dimension, two in this case. The remaining parameters, $\sigma_{ik}(i=1,\ldots,N, k=1,\ldots,K+1)$, ρ_i (where appropriate), p_{1_i}, and the initial conditions NIG_{0_i} or initial residuals $\eta_{i1}(i=1,\ldots,N)$, are transformed appropriately and assumed to arise from normal population distributions. Except for the initial residuals, these population distributions have unknown means and log-standard deviations with vague normal priors. The population mean initial residual is assumed to be zero and the population standard deviation is assigned an informative uniform prior on (0, 0.5), where the upper bound ensures that initial residuals greater than one are unlikely. The transformations applied are logarithmic for the residual standard deviations, initial conditions and p_1 parameters, and logistic for the ρ_i parameters (no transformation is required for the η_{i1}s).

3.5. Implementation issues

Dynamic models in BUGS are 'packaged' in one of two ways [33]. One option is to specify the differential equations using the BUGS language and pass these as arguments to a generic ordinary differential equation (ODE) solver. The alternative is to edit and compile a template module for 'hard-wiring' the ODE system into the software. In so doing, we create a new logical function in the BUGS language, which provides access to the numerical solution. We pass any parameters required to define the ODE system as arguments to the new function.

BUGS relies heavily on graphical modelling theory [29, 30]. However, it is important to note that in graphical modelling terms, forcing functions are non-standard nodes. At first glance we might think they are logical nodes, since they are deterministic in nature. However, logical nodes are deterministic functions of other nodes in the graph: when they are defined in terms of 'time' they are functions of *specific* times, whereas forcing functions are defined for *all* times, in particular, they are functions of the dummy variable of integration, which, technically speaking, does not belong in the graph (Figure 2). In cases where the ODE-system is defined using the BUGS language, however, the integration-dummy is accessible within the model description, since it is required to define the differential equations. Hence, we can also use this to define forcing functions, linearly interpolating between the 'forcing data' via the BUGS language. One very significant advantage of hard-wiring the ODE-system, however, is that it is potentially much faster to compute. In this case, the integration-dummy is only available within the hard-wired module, and so we need to pass the forcing data, along with the times to which they relate, as parameters to the new function, and perform the interpolation within—see the Appendix.

4. Results

We begin by fitting the basic model given by (5), with the individual-specific parameters, p_{1_i}, NIG_{0_i}, and σ_i, assumed to arise from log-normal population distributions with unknown means and standard deviations assigned vague normal and log-normal priors, respectively. We run 10 000 iterations in WinBUGS 1.4.3 [31, 32] with *WBDiff* interface [33] installed. WinBUGS code for the main models considered in this paper is given in the Appendix.

Primarily, we work with 'hard-wired' systems of equations by writing and compiling specialized BUGS-modules, but, in this case, we also considered specifying the model entirely via the BUGS language. The former took around 5 min on a 2.13-GHz machine (when coded efficiently—see the Appendix), whereas the latter approach was substantially slower, taking ~50 min. Point and interval estimates for the population parameters are presented in Table I, and a typical model fit is shown in Figure 1(b)—individual '10' was chosen as their data best illustrate the incremental benefit of increasing the model complexity.

Visual inspection of the model fits confirms that they are generally poor. Of primary importance in this paper is our ability to predict new data, and, to this end, the basic model is clearly inadequate. We might still wonder, however, whether it provides meaningful parameter estimates. From a clinical perspective, we are interested in the time delay that exists between glucose appearing in the blood and it then showing up on the sensor. This can be seen by looking at the relative positions of BG and SG peaks and/or troughs in Figure 1(a). The time delay is represented in the model by $\tau = p_1^{-1}$. To

Table I. Posterior median point estimates for population parameters (mean and SD), with 95 per cent credible intervals in parentheses, from analysis of Guardian® RT SG-BG data using three different models.

	Basic model		Calibrated model		Calibrated model + AR	
Parameter	Pop. mean (95 per cent CI)	Pop. SD (95 per cent CI)	Pop. mean (95 per cent CI)	Pop. SD (95 per cent CI)	Pop. mean (95 per cent CI)	Pop. SD (95 per cent CI)
$\log p_1$	−3.58 (−4.14, −3.03)	0.889 (0.597, 1.52)	−2.79 (−2.89, −2.67)	0.164 (0.102, 0.283)	−2.82 (−2.94, −2.71)	0.166 (0.0933, 0.301)
$\log F$	—	—	−0.198 (−0.291, −0.108)	0.316 (0.258, 0.396)	−0.202 (−0.289, −0.118)	0.298 (0.245, 0.370)
B	—	—	1.52 (0.981, 2.06)	1.76 (1.41, 2.24)	1.63 (1.19, 2.06)	1.37 (1.08, 1.83)
$\log \sigma$	−0.130 (−0.329, 0.0707)	0.320 (0.216, 0.537)	−1.42 (−1.60, −1.24)	0.615 (0.492, 0.782)	−2.14 (−2.27, −2.01)	0.445 (0.357, 0.564)
$\log NIG_0$	2.31 (2.01, 2.60)	0.458 (0.310, 0.750)	2.20 (1.83, 2.56)	0.588 (0.395, 0.980)	—	—
$\eta_{.1}$	—	—	—	—	0	0.374 (0.0413, 0.496)
ρ	—	—	—	—	0.8 (0.8, 0.8)	—

get a rough idea of its size purely from the data, we performed a crude correlation analysis in which the SG values were lagged by 0, 5, 10 min, etc., and the correlation between BG and lagged-SG was calculated. The largest correlation coefficient (0.937) was obtained at a lag of 15 min, suggesting that a model-based estimate of the population median delay (from Table I) of $\exp(3.58) \approx 36$ min could be somewhat inaccurate.

To fit the 'calibrated' model given by (6) with $\eta_{ij} \sim \mathrm{N}(0, \sigma^2_{iP[i,j]})$, we need to choose between the available exchangeability assumptions for the calibration parameters C_{ik}, $i=1, \ldots, N$, $k=1, \ldots, K+1$. In particular, we wish to explore whether or not the calibration parameters reflect characteristics of the individual (and/or sensor), that is, whether to include individual-level means for these parameters. To address this we fit the model with and without individual means, and assess performance by looking at the posterior mean deviance, as a measure of model fit, and the Deviance Information Criterion, which penalizes the former by adding a penalty equal to the 'effective number of parameters' [44]. We find that there is no support for individual-level means as the mean deviance is virtually the same for both models, regardless of the effective number of parameters. Hence, we proceed with a three-level, as opposed to a four-level, model.

We ran 300 000 iterations of WinBUGS for the three-level calibrated model in a little over 3.5 h. To reduce the amount of computer memory required to store the output, we retained only every fifth sample for each parameter. The final 50 000 of the resulting 60 000 samples were then used for inference. The increased run-length here is due to a high level of autocorrelation in the output for the newly introduced C_{ik} vectors. Parameter estimates for the population parameters are shown in the fourth and fifth columns of Table I. Now the population median time delay is 16.3 min, which is consistent with our crude empirical estimate. The inter-individual variability is relatively low, corresponding to a coefficient of variation, on the time-delay scale, of around 18 per cent. This is most usefully expressed, however, in the form of a prediction interval for new individuals' time delays, which accounts for uncertainty in the population mean and variability estimates: a 95 per cent interval is given by (11.1, 23.6) min. Population medians and 95 per cent prediction intervals for the calibration shift and scale factor are 1.52 $(-2.02, 5.09)$ mmol/L and 0.821 (0.431, 1.55), respectively. Note that the population median residual standard deviation has reduced from $\exp(-0.130) = 0.878$ to $\exp(-1.42) = 0.242$, indicating a substantially better fit to the data, as illustrated in Figure 1(c).

Although the model fit is much improved, we would still like, for prediction purposes, to be able to track the considerable fluctuations away from such fits that are apparent in the data. As can be seen in Figure 1(c) the residuals are serially correlated, and so we attempt to model them via the autoregressive (AR) process (7). However, a problem arises when we run the MCMC simulation. Recall that there is no penalty for large η_{ij}s, only their stochastic components γ_{ij} (see (7)) need to be small. A good fit to the data, then, can often be obtained by setting $\rho_i = 1$ and choosing other parameters such that the underlying 'model fit' $\{CIG_{ij}, j=1, \ldots, n\}$ lies a roughly constant (with respect to time) distance away from the data. Then the residuals are all similar, consistent with $\rho_i = 1$, and each requires only a small stochastic component. We have some control over this phenomenon in specifying an informative prior for the initial residual η_{i1}. However, it still occurs for some individuals unless we constrain the value of ρ, and while the resulting model fits well in terms of $\{\phi_{ij}, j=1, \ldots, n\}$, the underlying CIG_{ij} values are often implausible.

To impose the required constraint we assume that $\mathrm{logit}(\rho_i/\rho_{\max})$, as opposed to $\mathrm{logit}(\rho_i)$, arises from some normal population distribution (with unknown parameters), where $\rho_{\max}$ is assigned a specific value. However, now all of the individual ρ_is are estimated equal to $\rho_{\max}$, whatever value of $\rho_{\max}$ we choose. We address this by choosing the maximum value possible (in increments of 0.05) that still leads to plausible CIG-series for all individuals. Note that in so doing we find that there is no support for individual-specific ρ_is, and so we also set $\rho_i = \rho \; \forall i$, where $\mathrm{logit}(\rho/\rho_{\max})$ is assigned a vague normal prior. The value chosen for $\rho_{\max}$ was 0.8, and WinBUGS was again run for 300 000 iterations, retaining only every 5th sample. This took around 11.5 h, and point and interval estimates for the population parameters from the final 50 000 samples are presented in Table I. These are in good agreement with results from the previous model (without the AR process). The calibration shift B seems a little higher with less variability but we would expect from these figures that the underlying CIG-series are similar to before, as illustrated in Figure 3(b) for individual 10. Note that the population median residual standard deviation is now around half its previous value, at 0.118. This corresponds to the stochastic component of the residuals $\gamma_{ij} = \mathrm{SG}_{ij} - \phi_{ij}$, indicating that the ϕ-series offer a substantial improvement over the CIG-series, as we would hope, and as is demonstrated in Figure 3(a) for individual 10. To demonstrate

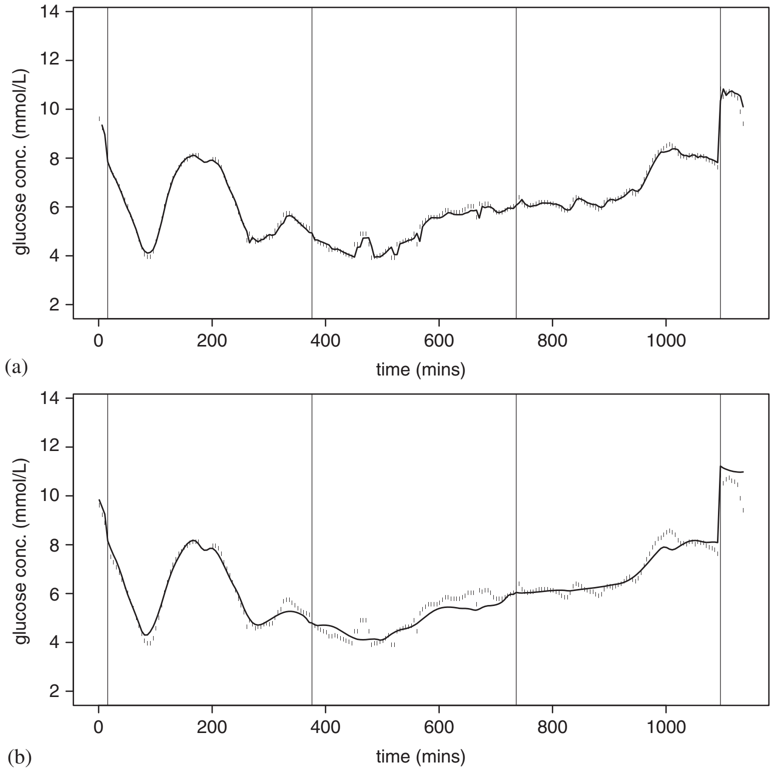

Figure 3. Observed SG data (|) and posterior median model-predicted concentrations for individual ‘10’ plotted against time since beginning of study: (a) ϕ_{10j} (—) from ‘calibrated + AR’ population model; and (b) CIG_{10j} (—) from ‘calibrated + AR’ population model.

the performance of our model across all individuals we present relative residuals, $100\times(\mathrm{SG}_{ij}-\phi_{ij})/\phi_{ij}$, for $i=1,\ldots,N=12$, $j=1,\ldots,n=228$, in Figure 4(a). (Relative residuals are chosen in preference to the γ_{ij}s as the percentage scale, on which they are defined, is more intuitive.) Ninety five per cent of all relative residuals have magnitudes less than 4.2 per cent, whereas 80 per cent are smaller than 2.1 per cent. The median size is 0.93 per cent. Note that individual 10 is fairly representative; if anything, he/she is one of the less well-fitted individuals. To further demonstrate the impact of the AR component in our model, we also present, in Figure 4(b), relative residuals for the case in which it is not present. Here the median relative residual size is 2.3 per cent, whereas 95 per cent of residuals have sizes below 8.9 per cent, and 80 per cent have sizes below 4.9 per cent. It is of interest to examine whether residuals corresponding to hypoglycaemic ($\leqslant 3.9$ mmol/L), euglycaemic (>3.9 mmol/L, $\leqslant 10$ mmol/L), and hyperglycaemic (>10 mmol/L) glucose ranges are similar or not. Some small differences are apparent, with median residual sizes (inter-quartile intervals in parentheses) for the three groups given by 0.241 (0.144, 0.279) mmol/L, 0.0650 (0.0280, 0.124) mmol/L, and 0.100 (0.0400, 0.190) mmol/L, respectively. (Note that the hypoglycaemic figures are based on only 15 residuals.) Visual inspection of various plots (not shown), including histograms of residual size for each group, and plots of all residuals versus the corresponding glucose values, reveal only small trends, however, and suggest that modifying the model for the residual variance would be of little practical benefit.

Finally, we wish to acknowledge some uncertainty regarding our choice of upper bound $\rho_{\max}$. Choosing a value for $\rho_{\max}$, it seems, is tantamount to fixing ρ at that value. We would like to acknowledge that ρ could lie between 0.75, which gives inferior model fits $\{\phi_{ij}, j=1,\ldots,n\}$, and 0.85, which leads to implausible CIG-series. But there is no point trying to acknowledge this uncertainty via a prior distribution, since we know that the posterior will be concentrated on the upper boundary (we may as well set $\rho=0.85$). Instead we specify ρ as a ‘distributional constant’—a fixed *distribution* as opposed to a fixed value (see [45], for example). We specify $\rho\sim\mathrm{Unif}(0.75, 0.85)$ but we prevent learning about ρ from the likelihood—in graphical modelling terms, ρ acts as a parent of $\{\mathrm{SG}_{ij}, i=1,\ldots,N, j=1,\ldots,n\}$ but the SG_{ij}s are not considered to be children of ρ. WinBUGS code for ‘cutting feedback’ from the likelihood in this way is presented in the Appendix, and further comments on the use of such

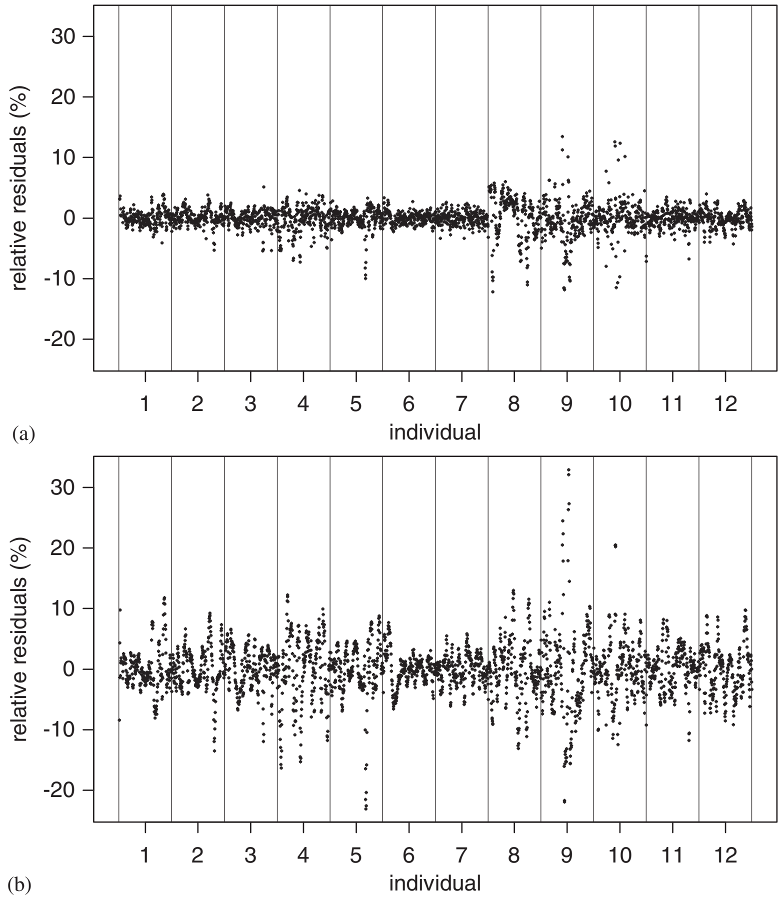

Figure 4. Relative residuals, $100\times(\mathrm{SG}_{ij}-\phi_{ij})/\phi_{ij}$, for all 12 individuals. Residuals are plotted in time-order within each individual's zone: (a) 'calibrated + AR' population model; and (b) 'calibrated' population model.

techniques are given in the discussion below. Results from this model give point estimates for population parameters identical to those obtained with $\rho_{\max}=0.8$, modulo Monte Carlo error, except for a very slight difference in the population median σ, which is still given by 0.118 to three significant figures.

5. Discussion

We have characterized the relationship between BG and SG, in children and adolescents, for the Guardian® RT CGM system. Various hierarchical models have been explored to determine the most appropriate model, which assumes that calibration parameters for the same individual/sensor are not correlated. Our model can be viewed as a hierarchical, nonlinear regression, where the regression function is given by numerically solving a differential equation with accompanying initial condition, forcing function, and unknown parameter. While the forcing function is not naturally accommodated within the BUGS software, due to it representing a new class of graphical node, we have illustrated how it can be incorporated.

Even without modelling autocorrelation among the residuals our model fits remarkably well, given its simplicity. However, there are clear, unexplained fluctuations away from such fits, which we have accounted for by simultaneously fitting the nonlinear regression and an AR(1) process for the residuals.[§] While this has presented several practical challenges, we emphasize that the resulting model fits are very satisfactory (the majority of residuals—nearly 80 per cent—correspond to percentage differences between data and model fit of less than 2 per cent), and predictions from the model faithfully

[§] The order of the AR process was chosen by fitting individual sets of posterior mean residuals using the 'ar' package in *R*.

reproduce features present in the observed data. *Sudden* deviations away from the fitted curve, which are more prominent with other CGM systems, might be better handled by assuming heavier tailed, t-distributed stochastic residuals (γ_{ij}) in the AR process, but we have not explored this yet. Note that Breton and Kovatchev [28] find the (unbounded) Johnson family of distributions [46] useful for the FreeStyle Navigator$^{\text{TM}}$ system. Another possible modification to the model for the residuals is to allow for different behaviour in hypoglycaemic, euglycaemic, and hyperglycaemic glucose ranges. However, any differences that could reliably be ascertained from our analyses were small, and it was thought that adapting the model would be of little practical benefit, especially considering that increasing the complexity of the residual model may exacerbate the apparent identifiability problem.

The motivation behind this work is to accelerate artificial pancreas development by providing a means of simulating large quantities of *realistic* sensor data, which can be fed into the control algorithm to test its response [17]. To this end, it is important to account for all sources of variability in glucose-sensor data. In particular, we have estimated the population distribution of regression parameters, including their means/medians, their inter- and intra-individual variabilities, and any correlations that may exist between parameters. This allows us to generate 'virtual patients' by simulating realistic parameter sets from the population distribution. In addition, the Bayesian nature of our model enables full acknowledgement of the uncertainty associated with each population parameter estimate when simulating the virtual patients. Each virtual patient's parameters can be input, along with observed or simulated BG data, into the nonlinear regression function to derive CIG-values. However, the regression function is parsimonious and simply adding Gaussian white noise to this is insufficient for the purposes of generating realistic sensor data. Hence, characterizing the residuals has been a vital step in our analyses. The resulting parameter estimates can be used, again fully acknowledging their uncertainty, to simulate AR processes to be added to the derived CIG-series.

It is somewhat disappointing that we have had to constrain the degree of autocorrelation ρ in the AR process and that the posterior distribution for ρ is then always concentrated on the artificial boundary. This is mainly due, we think, to an inability to encapsulate within our prior distribution our 'common sense' knowledge as to the relationship between the observed data and the model-predicted CIG-series. Basically, we believe that the CIG-series should fit the data reasonably well and that the AR process should then account for mild fluctuations around that fit. But the unconstrained posterior is located near $\rho = 1$ with CIG-series that often lie, implausibly, some (roughly) constant distance from the data. The resulting individual-level parameters, B, F, and p_1 are clearly inappropriate for the given individual, but do not necessarily have outlying values in terms of the population distribution of those parameters. Hence, it is not possible to circumvent the problem by constraining B, F, and p_1. We have to remember that our model represents a gross simplification of the underlying process. To put too much emphasis on fitting the data when the model is known to be 'wrong' would be a mistake, in our opinion—the model is designed to provide meaningful parameter estimates and reasonably realistic predictions, which we believe it achieves.

One approach to avoiding the confounding problem would be to perform the analysis in two stages. We could first fit the calibrated model without the AR process and use this for inference on the parameters. We could then apply an autoregressive model to the residuals in order to characterize any fluctuations away from the deterministic model. However, this would ignore uncertainty in the model fit and prevent the model fit being adjusted, even slightly, to accommodate the different error structure. Hence, our efforts to fit nonlinear regression and AR models simultaneously. However, fixing $\rho_{\max}$ is tantamount to fixing ρ and we would prefer to acknowledge some uncertainty regarding the latter. We achieve this by specifying ρ as a 'distributional constant', as opposed to a fixed value, by placing a valve in the graphical model that allows information to flow from prior to likelihood but not vice versa. This allows us to be uncertain about ρ without the model fit being 'tweaked' inappropriately. Point estimates provided by this approach are identical to those obtained with $\rho_{\max} = 0.8$, but we would prefer to use the former for prediction as the level of uncertainty would be more realistic. Cutting the feedback from one or more sources of likelihood in such a way is growing in popularity—see, for example, [45, 47–54]. The motives vary, but it is typically used for 'multiply imputing' missing data or combining different sub-models that might otherwise be somewhat inconsistent, due to misspecification, say.

Other ways to cut feedback in the model begin with duplicating the SG data. A homoscedastic model could be specified for the first set of data and the same B_{ik}, F_{ik}, and p_{1_i} parameters could then be used to define the CIG$_{ij}$s needed for fitting an AR model to the second set of data. Without appropriate valves/cuts in the graph, this would lead to excessive precision due to using the data twice. If we cut the feedback from the second set of data to the B_{ik}, F_{ik}, and p_{1_i} parameters, however, then we are guaranteed plausible CIG-series regardless of what happens to ρ. Note that ρ can then be unconstrained—the posterior median and 95 per cent credible interval from our analyses are 0.925 and (0.908, 0.941), respectively (results for other parameters are the same as in columns 4 and 5 of Table I). This is a Bayesian analogue of the two-stage approach described above where uncertainty in the model fit is now acknowledged. Further research is required to address whether or not this approach is preferable to specifying ρ as a distributional constant. Another option would be to also create duplicate B_{ik}, F_{ik}, and p_{1_i} parameters, a set for each model fit, and to cut feedback from the AR set to the common population parameters (see [45], for example). This does not help for our data, however, as the parameters for the AR fit become implausible with unconstrained ρ.

One area that we have not considered in this paper is uncertainty in the forcing function. Here we have simply linearly interpolated between a series of observed values, but it is natural to think that those observations might be subject to some error, resulting in a somewhat jagged forcing function. This is likely at odds with our prior beliefs as we might expect a biological process to be largely smooth. Moreover, if there is noise in the time-series then, presumably, we would rather use the underlying ‘true’ values. Hence, we may wish to specify a separate sub-model for the BG data. This is the subject of ongoing research and will form the basis of a future report. Another methodological issue is the numerical stability of ODE solving algorithms, which can be sensitive to the values of the input parameters, in some more complex settings precluding the use of vague prior distributions, say. The works of Ramsay *et al.* [55] and Campbell [56] offer an alternative approach that circumvents this problem. However, it is, as yet, unclear to us how this might be implemented in a flexible modelling framework such as BUGS or JAGS [57]. More robust solving algorithms for BUGS are currently under investigation.

Appendix A: WinBUGS/WBDiff code

WinBUGS code for the calibrated model (without AR errors) is presented below. Most of the code is self-explanatory but some notes, pertaining to the line numbers given in the right-hand margin, are provided below for clarity.

```
model {
   for (i in 1:N) {                                                    #1
      # likelihood...                                                  #2
      for (j in 1:n) {                                                 #3
         CIG[i,j] <- F[i, P[i,j]]*NIG[i,j] + B[i, P[i,j]]              #4
      }                                                                #5
      for (j in 1:n) {                                                 #6
         SG[i,j] ~ dnorm(CIG[i,j], prec[i,j])                          #7
         prec[i,j] <- 1/pow(sigma[i, P[i,j]], 2)                       #8
      }                                                                #9
      NIG[i,1:n] <- glucose(init[i], t[i,1:n], par[i,1:n.par],
                           t[i,1], tol)                                #10
      for (l in 1:m) {                                                 #11
         par[i,l] <- s[l]                                              #12
         par[i,m+l] <- BG[i,l]                                         #13
      }                                                                #14
      par[i,2*m+1] <- p1[i]                                            #15
                                                                       #16
      # exchangeability assumptions...                                 #17
      log(init[i]) <- log.init[i]                                      #18
      log.init[i] ~ dnorm(mean.init, prec.init)                        #19
      log(p1[i]) <- log.p1[i]                                          #20
      log.p1[i] ~ dnorm(mean.p1, prec.p1)                              #21
      for (j in 1:5) {                                                 #22
         log(sigma[i,j]) <- log.sigma[i,j]                             #23
         log.sigma[i,j] ~ dnorm(mean.sigma, prec.sigma)                #24
         log(F[i,j]) <- C[i,j,1]                                       #25
      B[i,j] <- C[i,j,2]                                          #26
      C[i,j, 1:2] ~ dmnorm(mean.C[], prec.C[,])                   #27
    }                                                             #28
  }                                                               #29
                                                                  #30
  # priors...                                                     #31
  mean.init ~ dnorm(0, 0.0001)                                    #32
  prec.init <- 1/pow(sd.init, 2)                                  #33
  log(sd.init) <- log.sd.init                                     #34
  log.sd.init ~ dnorm(0, 0.0001)                                  #35
                                                                  #36
  mean.sigma ~ dnorm(0, 0.0001)                                   #37
  prec.sigma <- 1/pow(sd.sigma, 2)                                #38
  log(sd.sigma) <- log.sd.sigma                                   #39
  log.sd.sigma ~ dnorm(0, 0.0001)                                 #40
                                                                  #41
  mean.p1 ~ dnorm(0, 0.0001)                                      #42
  prec.p1 <- 1/pow(sd.p1, 2)                                      #43
  log(sd.p1) <- log.sd.p1                                         #44
  log.sd.p1 ~ dnorm(0, 0.0001)                                    #45
                                                                  #46
  mean.C[1:2] ~ dmnorm(zero[], prior.prec.C[,])                   #47
  prec.C[1:2,1:2] ~ dwish(prior.mat.C[,], 2)                      #48
  cov.C[1:2,1:2] <- inverse(prec.C[,])                            #49
}
```

- Line 4: The `P[,]` variable, representing the calibration period to which each observation belongs, is fixed as the calibration times are assumed known. Hence, it can be defined in the data set as opposed to being calculated in the BUGS code.
- Lines 7–8: Normal distributions in BUGS are parameterized in terms of mean and precision (1/variance). Hence, $\texttt{prec[i,j]} = 1/\sigma^2_{i\,P[i,j]}$.
- Line 10: `glucose(.)` is the name given to our new 'hard-wired' function that specifies and solves the differential equation (4) at a specified grid of time-points. It is a vector-valued function of five arguments: (i) the initial condition for individual i, NIG_{0_i}, denoted `init[i]`; (ii) the vector of times at which the solution is to be evaluated; (iii) a set of parameters required to fully specify the differential equation—these are defined on lines 11–15 (see below); (iv) the time to which the initial condition applies; and (v) the numerical tolerance to be used by the solving algorithm in determining whether or not the solution is sufficiently accurate—a value of 10^{-6} is typical. Pseudo-code for the hard-wired function is given in Appendix B.
- Lines 11–15: In order that our new, hard-wired component can evaluate the forcing function, we must supply it with the forcing data, and the times to which they relate. More generally, we may also need to supply the *number* of forcing data but here this is the same for all individuals, and so this information can be hard-wired instead of being passed as a parameter. The only other parameter required to fully specify the differential equation is p_{1_i}.
- Line 19 onwards: Throughout, `mean.x` and `prec.x` denote the unknown population mean and precision of appropriately transformed `x`.
- Lines 47–48: `zero[]`, `prior.prec.C[,]`, and `prior.mat.C[,]` are specified in the data set: they are given by a two-dimensional vector of zeros, $0.0001 \times I_2$, and I_2, respectively, where I_2 denotes the 2×2 identity matrix.

The above model is adapted to incorporate autoregressive errors (with a common autocorrelation parameter ρ for all individuals) by first making the `j`-loop on line 6 run from 2 to `n` rather than 1 to `n` and by replacing `CIG[i,j]` on line 7 with `phi[i,j]`. The following code is then inserted into that `j`-loop.

```
    phi[i,j] <- CIG[i,j] + rho*eta[i,j-1]*F[i, P[i,j]]/F[i, P[i,j-1]]
    eta[i,j] <- SG[i,j] - CIG[i,j]
```

In addition, lines 18–19 above are replaced by

```
eta[i,1] ~ dnorm(0, prec.eta)
init[i] <- (SG[i,1] - eta[i,1] - B[i,1])/F[i,1]
```

and lines 32–35 are replaced by the blocks of code labelled `A` and `B` as follows:

```
# block A...
rho <- rho.max * prop
logit(prop) <- r
r ~ dnorm(0, 0.0001)
# block B...
prec.eta <- 1/pow(sd.eta, 2)
sd.eta ~ dunif(0, 0.5)
```

If we wish to specify `rho` as a 'distributional constant' instead, then we may replace block `A` with, for example,

```
rho <- cut(rho.dc)
rho.dc ~ dunif(0.75, 0.85)
```

Here the `cut(.)` function makes a 'copy' of the argument `rho.dc`. This always has the same value as the argument, and so uncertainty is propagated into the graph, but the 'cut' acts as a valve in the graph, preventing information from flowing through `rho` in the other direction (back to `rho.dc`). Hence, feedback from the likelihood is prevented and no learning about `rho.dc` can occur.

Appendix B: Pseudo-code for hard-wired differential equation with forcing function

Here we present pseudo-code for evaluating the differential equation(s) in our 'hard-wired' module. (The reader is referred to the WBDiff documentation [33] for more general information.) The input parameters, including the forcing data, are available via a vector named `theta` and we are required to evaluate the equation(s) at some arbitrary time `t`. Note that the elements of `theta` are arranged as defined by the BUGS code in Appendix A, i.e. m forcing times, followed by m forcing data, followed by p_1. Note also that the current solution at time `t` is also made available by the solving algorithm—this is denoted by `NIG` in the pseudo-code below. The main difficulty is in finding which elements of the forcing data are relevant to time `t`, i.e. which successive pair of observations lie either side of `t`, so that we can interpolate between them. We should be aware that the code will be called many, many times in solving the equation(s), and so it may be prudent to think about efficiency. Here we realize that each time the code is called, there is a very good chance that the 'forcing interval' will be the same as that most recently used or the subsequent interval. Hence, each time we evaluate the equation(s), we store, in a global variable named `prev`, the index of `theta` corresponding to the end of the forcing interval, so that we can refer to it next time.

```
IF (t >= theta[prev-1]) & (t < theta[prev]) THEN
  end = prev
ELSE IF (prev < m) & (t >= theta[prev]) & (t < theta[prev+1]) THEN
  end = prev + 1
ELSE
  find end such that (t >= theta[end-1]) & (t < theta[end])
END
prev = end
gradient = (theta[m+end] - theta[m+end-1])/(theta[end] - theta[end-1])
BG = theta[m+end-1] + gradient*(t - theta[end-1])
RETURN theta[2*m+1]*(BG - NIG)
```

Acknowledgements

D. J. L and C. W. are funded by the UK Medical Research Council (grant code U.1052.00.005). R. H. was supported by the Juvenile Diabetes Research Foundation (grants 22-2006-1113 and 22-2007-1801) and the

National Institute for Health Research Cambridge Biomedical Research Centre. They are grateful to Martyn Plummer and David Spiegelhalter for helpful discussions, and to one referee, whose comments helped them to improve on an earlier version of this paper.

References

1. Atkinson MA, Eisenbarth GS. Type 1 diabetes: new perspectives on disease pathogenesis and treatment. *The Lancet* 2001; **358**:221–229.
2. Diabetic Control and Complications Trial Research Group. The effect of intensive treatment of diabetes on the development and progression of long term complications in insulin-dependent diabetes mellitus. *New England Journal of Medicine* 1993; **329**:977–986.
3. American Diabetes Association. Standards of medical care in diabetes—2010. *Diabetes Care* 2010; **33**:S11–S96.
4. Pickup J, Keen H. Continuous subcutaneous insulin infusion at 25 years: evidence base for the expanding use of insulin pump therapy in type 1 diabetes. *Diabetes Care* 2002; **25**:593–598.
5. Cryer PE. The barrier of hypoglycemia in diabetes. *Diabetes* 2008; **57**:3169–3176.
6. Heinemann L. Variability of insulin absorption and insulin action. *Diabetes Technology and Therapeutics* 2002; **4**:673–682.
7. Carroll MF, Schade DS. The dawn phenomenon revisited: implications for diabetes therapy. *Endocrine Practice* 2005; **11**:55–64.
8. Riazi A, Pickup J, Bradley C. Daily stress and glycaemic control in Type 1 diabetes: individual differences in magnitude, direction, and timing of stress-reactivity. *Diabetes Research and Clinical Practice* 2004; **66**:237–244.
9. Klonoff DC. Continuous glucose monitoring: roadmap for 21st century diabetes therapy. *Diabetes Care* 2005; **28**: 1231–1239.
10. Kamath A, Mahalingam A, Brauker J. Methods of evaluating the utility of continuous glucose monitor alerts. *Journal of Diabetes Science and Technology* 2010; **4**:57–66.
11. Tamborlane WV, Beck RW, Bode BW, Buckingham B, Chase P, Clemons R, Fiallo-Scharer R, Fox LA, Gilliam LK, Hirsch IB, Huang ES, Kollman C, Kowalski AJ, Laffel L, Lawrence JM, Lee J, Mauras N, O'Grady M, Ruedy KJ, Tansey M, Tsalikian E, Weinzimer S, Wilson DM, Wolpert H, Wysocki T, Xing D. Continuous glucose monitoring and intensive treatment of type 1 diabetes. *New England Journal of Medicine* 2008; **359**:1464–1476.
12. Juvenile Diabetes Research Foundation Continuous Glucose Monitoring Study Group. The effect of continuous glucose monitoring in well-controlled type 1 diabetes. *Diabetes Care* 2009; **32**:1378–1383.
13. Hovorka R. Continuous glucose monitoring and closed-loop systems. *Diabetic Medicine* 2006; **23**:1–12.
14. Steil GM, Rebrin K, Hariri F, Jinagonda S, Tadros S, Darwin C, Saad MF. Interstitial fluid glucose dynamics during insulin-induced hypoglycaemia. *Diabetologia* 2005; **48**:1833–1840.
15. Pfeiffer EF, Meyerhoff C, Bischof F, Keck FS, Kerner W. On line continuous monitoring of subcutaneous tissue glucose is feasible by combining portable glucosensor with microdialysis. *Hormone and Metabolic Research* 1993; **25**:121–124.
16. Sternberg F, Meyerhoff C, Mennel FJ, Mayer H, Bischof F, Pfeiffer EF. Does fall in tissue glucose precede fall in blood glucose? *Diabetologia* 1996; **39**:609–612.
17. Wilinska ME, Chassin LJ, Acerini CL, Allen JM, Dunger DB, Hovorka R. Simulation environment to evaluate closed-loop insulin delivery systems in type 1 diabetes. *Journal of Diabetes Science and Technology* 2010; **4**:132–144.
18. Diabetes Research in Children Network (DirecNet) Study Group. The accuracy of the CGMS in children with type 1 diabetes: results of the Diabetes Research in Children Network (DirecNet) accuracy study. *Diabetes Technology and Therapeutics* 2003; **5**:781–789.
19. Diabetes Research in Children Network (DirecNet) Study Group. The accuracy of the GlucoWatch G2 biographer in children with type 1 diabetes: results of the Diabetes Research in Children Nnetwork (DirecNet) accuracy study. *Diabetes Technology and Therapeutics* 2003; **5**:791–800.
20. Kovatchev B, Anderson S, Heinemann L, Clarke W. Comparison of the numerical and clinical accuracy of four continuous glucose monitors. *Diabetes Care* 2008; **31**:1160–1164.
21. Keenan DB, Cartaya R, Mastrototaro JJ. Accuracy of a new real-time continuous glucose monitoring algorithm. *Journal of Diabetes Science and Technology* 2010; **4**:111–118.
22. Kamath A, Mahalingam A, Brauker J. Analysis of time lags and other sources of error of the DexCom SEVEN continuous glucose monitor. *Diabetes Technology and Therapeutics* 2009; **11**:689–695.
23. Beck RW, Buckingham B, Miller K, Wolpert H, Xing D, Block JM, Chase HP, Hirsch I, Kollman C, Laffel L, Lawrence JM, Milaszewski K, Ruedy KJ, Tamborlane WV. Factors predictive of use and of benefit from continuous glucose monitoring in type 1 diabetes. *Diabetes Care* 2009; **32**:1947–1953.
24. Hovorka R, Allen JM, Elleri D, Chassin LJ, Harris J, Xing D, Kollman C, Hovorka T, Larsen AM, Nodale M, De Palma A, Wilinska ME, Acerini CL, Dunger DB. Manual closed-loop insulin delivery in children and adolescents with type 1 diabetes: a phase 2 randomised crossover trial. *The Lancet* 2010; **375**:743–751.
25. Hovorka R, Wilinska ME, Chassin LJ, Dunger DB. Roadmap to the artificial pancreas. *Diabetes Research and Clinical Practice* 2006; **74**:S178–S182.
26. Bequette BW. A critical assessment of algorithms and challenges in the development of a closed-loop artificial pancreas. *Diabetes Technology and Therapeutics* 2005; **7**:28–47.
27. Bode B, Gross K, Rikalo N, Schwartz S, Wahl T, Page C, Gross T, Mastrototaro J. Alarms based on real-time sensor glucose values alert patients to hypo- and hyperglycemia: the Guardian continuous monitoring system. *Diabetes Technology and Therapeutics* 2004; **6**:105–113.
28. Breton M, Kovatchev B. Analysis, modeling, and simulation of the accuracy of continuous glucose sensors. *Journal of Diabetes Science and Technology* 2008; **2**:853–862.

29. Lauritzen SL, Dawid AP, Larsen BN, Leimer HG. Independence properties of directed Markov fields. *Networks* 1990; **20**:491–505.
30. Spiegelhalter DJ. Bayesian graphical modelling: a case-study in monitoring health outcomes. *Applied Statistics* 1998; **47**:115–133.
31. Lunn DJ, Thomas A, Best N, Spiegelhalter D. WinBUGS—a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and Computing* 2000; **10**:325–337.
32. Lunn D, Spiegelhalter D, Thomas A, Best N. The BUGS project: evolution, critique and future directions (with Discussion). *Statistics in Medicine* 2009; **28**:3049–3067.
33. Lunn D. WinBUGS Differential Interface—worked examples, Department of Epidemiology and Public Health, Imperial College School of Medicine, London, 2004.
34. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes in* C++: *The Art of Scientific Computing*. (2nd edn). Cambridge University Press: Cambridge, U.K., 2002; 719–727.
35. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equations of state calculations by fast computing machines. *Journal of Chemical Physics* 1953; **21**:1087–1091.
36. Hastings WK. Monte Carlo sampling-based methods using Markov chains and their applications. *Biometrika* 1970; **57**:97–109.
37. Jayne SR, Tokmakian R. Forcing and sampling of ocean general circulation models: impact of high-frequency motions. *Journal of Physical Oceanography* 1997; **27**:1173–1179.
38. Coffaro G, Sfriso A. Simulation model of Ulva rigida growth in shallow water of the Lagoon of Venice. *Ecological Modelling* 1997; **102**:55–66.
39. Sunehag AL, Dalla Man C, Toffolo G, Haymond MW, Bier DM, Cobelli C. Beta-cell function and insulin sensitivity in adolescents from an OGTT. *Obesity* 2009; **17**:233–239.
40. Rebrin K, Steil GM, van Antwerp WP, Mastrototaro JJ. Subcutaneous glucose predicts plasma glucose independent of insulin: implications for continuous monitoring. *American Journal of Physiology* 1999; **277**:E561–E571.
41. Wilinska ME, Bodenlenz M, Chassin LJ, Schaller HC, Schaupp LA, Pieber TR, Hovorka R. Interstitial glucose kinetics in subjects with type 1 diabetes under physiologic conditions. *Metabolism* 2004; **53**:1484–1491.
42. Besag J. Spatial interaction and the statistical analysis of lattice systems (with Discussion). *Journal of The Royal Statistical Society*, *Series B* 1974; **36**:192–236.
43. Besag J, York JC, Mollié A. Bayesian image restoration, with two applications in spatial statistics (with Discussion). *Annals of the Institute of Statistical Mathematics* 1991; **43**:1–59.
44. Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A. Bayesian measures of model complexity and fit (with Discussion). *Journal of The Royal Statistical Society*, *Series B* 2002; **64**:583–639.
45. Lunn D, Best N, Spiegelhalter D, Graham G, Neuenschwander B. Combining MCMC with 'sequential' PKPD modelling. *Journal of Pharmacokinetics and Pharmacodynamics* 2009; **36**:19–38.
46. Johnson N. Systems of frequency curves generated by methods of translation. *Biometrika* 1949; **36**:149–176.
47. Bhattacharjee M, Botting C, Sillanpää M. Bayesian biomarker identification based on marker-expression proteomics data. *Genomics* 2008; **92**:384–392.
48. Carrigan G, Barnett A, Dobson A, Mishra G. Compensating for missing data from longitudinal studies using WinBUGS. *Journal of Statistical Software* 2007; **19**(7).
49. Choi J, Fuentes M, Reich BJ. Spatial–temporal association between fine particulate matter and daily mortality. *Computational Statistics and Data Analysis* 2009; **53**:2989–3000.
50. He Y, Zaslavsky A. Combining information from cancer registry and medical records data to improve analyses of adjuvant cancer therapies. *Biometrics* 2009; **65**(3):946–952.
51. Jackson C, Best N, Richardson S. Bayesian graphical models for regression on multiple data sets with different variables. *Biostatistics* 2009; **10**:335–351.
52. Jackson C, Best N, Richardson S. Hierarchical related regression for combining aggregate and individual data in studies of socio-economic disease risk factors. *Journal of the Royal Statistical Society*, *Series A* 2008; **171**:159–178.
53. McCandless L, Douglas I, Evans S, Smeeth L. Cutting feedback in Bayesian regression adjustment for the propensity score. *The International Journal of Biostatistics* 2010; **6**(2). Article 16. DOI: 10.2202/1557–4679.1205.
54. Mwalili S, Lesaffre E, Declerck D. A Bayesian ordinal logistic regression model to correct for inter-observer measurement error in a geographical oral health study. *Journal of the Royal Statistical Society*, *Series C* 2005; **54**:77–93.
55. Ramsay JO, Hooker G, Campbell D, Cao J. Parameter estimation for differential equations: a generalized smoothing approach (with Discussion). *Journal of The Royal Statistical Society*, *Series B* 2007; **69**:741–796.
56. Campbell D. Bayesian collocation tempering and generalized profiling for estimation of parameters from differential equation models. *Ph.D. Thesis*, McGill University, Montreal, Canada, 2007.
57. Plummer M. JAGS Version 1.0.3 Manual. IARC, Lyon, France, 2008.