

# Fitting growth curve models in the Bayesian framework

Zita Oravecz<sup>1</sup> · Chelsea Muth<sup>2</sup>

Published online: 10 May 2017  
© Psychonomic Society, Inc. 2017

**Abstract** Growth curve modeling is a popular methodological tool due to its flexibility in simultaneously analyzing both within-person effects (e.g., assessing change over time for one person) and between-person effects (e.g., comparing differences in the change trajectories across people). This paper is a practical exposure to fitting growth curve models in the hierarchical Bayesian framework. First the mathematical formulation of growth curve models is provided. Then we give step-by-step guidelines on how to fit these models in the hierarchical Bayesian framework with corresponding computer scripts (JAGS and R). To illustrate the Bayesian GCM approach, we analyze a data set from a longitudinal study of marital relationship quality. We provide our computer code and example data set so that the reader can have hands-on experience fitting the growth curve model.

**Keywords** Bayesian modeling · Growth curve modeling

Longitudinal data are defined as repeated measures over time from the same unit (e.g., person). In psychology, longitudinal studies aim to measure certain characteristics of individuals, and their time scales span from short, such as one-day diurnal studies, to long, such as macro time scales

across a lifespan. Compared to cross-sectional research, which emphasizes differences between individuals, longitudinal research emphasizes changes within the individual. Measures of person-specific, or within-person change, allow longitudinal studies to capitalize on capturing how people differ from each other in their change patterns over time. The most frequently used longitudinal designs in psychology help us zoom in on the facets of individual change trajectories, and zoom out to compare multiple trajectories.

Longitudinal models are typically formulated in one of the following frameworks: ANOVA (see, e.g., in Laursen & Little, 2012), multilevel or hierarchical modeling (see, e.g., Raudenbush & Bryk, 2002), generalized linear mixed modeling (GLMM, see, e.g., Verbeke & Molenberghs, 2009), structural equation modeling (see, e.g., Bollen & Curran, 2006), dynamical systems modeling (see, e.g., Boker & Wenger, 2012) or Bayesian modeling (see, e.g., Gelman & Hill, 2006). It is important to note that the first 5 frameworks above determine model specification, but do not set a fixed estimation method; thus they can overlap with the Bayesian framework depending on the choice of estimation framework. Bayesian models most often involve Markov chain Monte Carlo algorithms for estimation (see more in Robert & Casella, 2004), while most other frameworks rely on maximum likelihood estimation (MLE). For example: multilevel and hierarchical models can be formulated in the GLMM framework, while a Bayesian version of these models can also be specified.

The main merit of longitudinal modeling is that it provides insights into mechanisms of within-person change, or intraindividual variability. This person-specific focus permits better understanding of basic human functioning and development, and addresses many interesting questions that arise in the field of psychology (see, e.g., Molenaar, 2004). Repeated measurements from one individual exhibit certain

---

✉ Zita Oravecz  
zita@psu.edu

Chelsea Muth  
muth.ca@gmail.com

<sup>1</sup> The Pennsylvania State University, 216 Health and Human Development Building, State College, PA, USA

<sup>2</sup> The Pennsylvania State University, 008 Health and Human Development Building, State College, PA, USA

similarities that are quantified by longitudinal models, which provide parameters for person-specific patterns and change mechanisms that cross-sectional approaches cannot capture. Moreover, focusing only on group-level characteristics and neglecting within-person trends can lead to erroneous conclusions as averaged curves can show a distorted representation of the individuals' growth curves (see, e.g., in Brown & Heathcote, 2003, also known as Simpson's paradox, see, e.g., Hamaker, 2012; Kievit, 2013).

In this paper we focus on one of the most frequently used classes of longitudinal models in psychology, the growth curve model (GCM). Depending on researcher's methodological framework, GCMs are also referred to as latent trajectory or latent curve models (see, e.g., Laursen & Little, 2003). The GCM allows us to model individual trajectories over time, and to compare these trajectories across individuals and groups. In application, the GCM has been used to assess a broad range of multifaceted longitudinal studies. To name a few, Curran and Bollen (2001) investigated the developmental link between antisocial behavior and depression; Mirman et al. (2008) modeled the time course of eye tracking data in a language processing task; Ferrer and McArdle (2004) tested new hypotheses about how cognitive ability from childhood to early adulthood influences academic achievement; Widaman and Reise (1997) examined measurement invariance of psychological constructs, and whether latent factors - such as attitudes towards smoking (i.e., drug abuse) - exhibit structural growth over time; and Walker et al. (2007) modeled change over time in moral reasoning development in children and adolescents. Good summaries of GCM applications can be found in books on longitudinal and latent variable modeling (e.g., Preacher, 2008; Bollen & Curran, 2006; Little, 2013; Singer & Willett, 2003).

Growth curve modeling is a popular methodological tool due to its flexibility in simultaneously analyzing both within-person (e.g., changes with age, change due to intervention, etc.) and between-person effects (i.e., individual differences); in other words, the GCM models inter-individual differences in intra-individual variation. A person-specific growth trajectory, specified as a mathematical function that describes how variables relate to each other over time, captures how an individual uniquely changes. This paper focuses on the linear growth curve model, in which the function of change is linear. Note that curvilinear polynomial functions (e.g., quadratic, cubic, etc.) also fall under the linear GCM umbrella - meaning that we are not limited to considering straight-line functional growth. Beyond handling varying growth functions, GCM can flexibly handle unbalanced designs, meaning study participants may be measured at different occasions and need not be excluded from analysis if some of their measurements are missing.

We advocate using the Bayesian statistical framework (see, e.g., Gelman et al., 2013; Kruschke, 2015; McElreath,

2016) for fitting growth curve models. Bayesian methods provide flexible tools to fit GCMs with various levels of complexity. We will show that by using generic Bayesian software programs, such as JAGS (see e.g., Plummer et al., 2003), a variety of simple and complex growth curve models can be implemented in a straightforward manner. One unique strength of Bayesian modeling is that results can be interpreted in an intuitive and direct way by summarizing the posterior distribution of the parameters. The posterior distribution can be described in terms of the likely range of parameters, and is derived based on the data without depending on hypothetical data or on the data collection plan. Moreover, prior knowledge on the likely values of the parameters can be included in the Bayesian framework, providing a principled approach to accumulating and incorporating scientific knowledge.

This paper introduces a basic growth curve model, provides a detailed description with computer code for fitting the model, and gives guidelines on how to incorporate further extensions for particular research questions. We published our computer code and example data set, described below, in a Git repository - available for the user via this link: <https://git.psu.edu/zzo1/FittingGCMBayesian> - to provide hands-on experience with model fitting.

To demonstrate the advantages of the Bayesian approach to GCM, we analyze a data set from a longitudinal study of marital relationship quality, measured across the transition to parenthood, described in Belsky and Rovine (1990). We are interested in studying how a father's feelings of marital love change in conjunction with the birth of his first child. We will show how to describe each father's change in experiencing marital love as a function of time (the love trajectory), beginning at 3 months before the birth of his first child. Moreover, to study the moderating effect of the fathers' overall positivity towards marriage, we will group fathers according to their number of positive experiences in marriage up to 3 months before birth.

We start by explaining the Bayesian formulations of the growth curve model. Next we provide details on the marital study. This is followed by a step-by-step guide on how to analyze the marital data with a Bayesian GCM. Then we consider several extensions that can easily be incorporated to extend the introduced GCM. Finally, we recap the potential applications and benefits of using this modeling framework.

### Capturing dependency in repeated measures via growth curve modeling in the Bayesian framework

Growth curve modeling can be seen as a multilevel regression technique, a special case of the generalized linear mixed model designed for analyzing the time course of one or more selected variables of interest. Below we describe the

GCM in the GLMM framework. Historically however, the GCM originates from confirmatory factor analysis. As such, a similar model can also be formulated in the structural equation modeling framework (see more details in Kaplan, 2014, page 220).

Within-person measurements are prone to show similarity. Therefore a major challenge of longitudinal statistical analysis is to adequately account for the association in measurements. Such association manifests in repeated measure clustering, meaning that measures exhibit person-specific, or within-person, variance patterns. These patterns violate statistical assumptions of the data being independent and identically distributed. In other words, if unaccounted for, this dependency leads to biased parameter estimates and underestimated standard errors (see e.g., in Diggle et al., 2002). The GCM, like any multilevel model, accounts for longitudinal dependency by adaptively modeling grouped data. Participants are considered exchangeable, and the measurement occasions within-person are ordered in time. The GCM assumes that each measurement has a noise component which follows a specific distribution, centered on the underlying growth curve. Moreover, these noise components are independent and unrelated each other. When the growth curve adequately captures the true underlying trend, the estimates will not be contaminated by systematic changes across time. Although this tutorial will focus on the simplest distribution (i.e., normal) for noise structure over time, we will give guidelines for extending this to model complex structures.

Though commonly specified in the classical frequentist framework, the GCM can be specified in the Bayesian statistical framework as well. In Bayesian models, parameters are conceptualized as random variables with probability distributions. Therefore we must assign a probability distribution for each model parameter before fitting the model. This distribution is called the prior distribution, and it expresses our knowledge about the most likely values of the unknown parameters. Note that these prior specifications are integral to Bayesian modeling, and should always be carefully chosen and justified when fitting a GCM in the Bayesian framework. For the current example we will set priors that extend broadly over the plausible range of the data, expressing vague information on the likely values of the parameters. The implication of a vague prior like this is that it has minimal influence on the quantitative details of the posterior distribution. Once priors are specified, we fit our Bayesian model by calculating the posterior distribution of model parameters: the product of the prior distributions and the data likelihood, normalized by the marginal likelihood. The posterior is the focal point of Bayesian estimation, representing the updated probability distribution assigned to our parameters after conditioning on our data. We will show how to calculate the posterior distribution for each model

parameter, that is, a probability distribution describing the most likely values of a GCM parameter given our data and other model parameters.

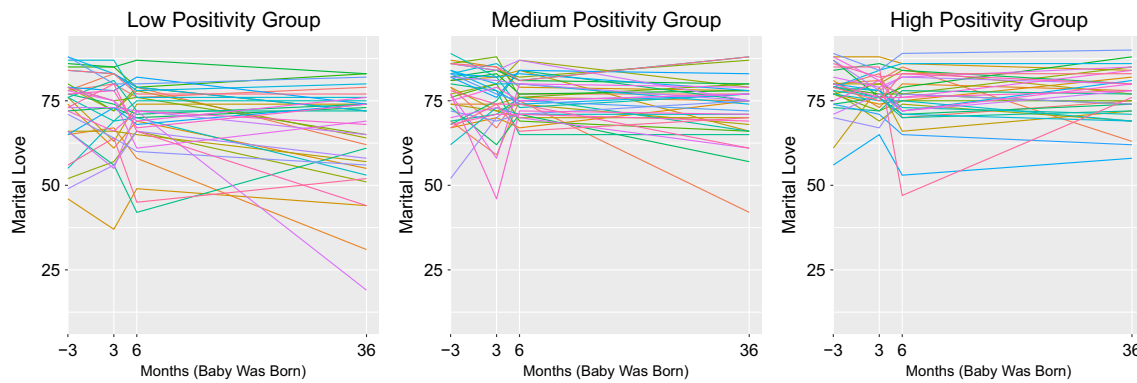
### The research question

Our illustration of the Bayesian GCM approach is based on a longitudinal marital love study ( $N = 106$ ), with 4 measurements per subject (father). The measurements aim to capture levels of marital love, quantified via self-reports on the Braiker and Kelley (1979) 9-point 10-item love subscale (see more details on the scales in the study description). Measurements were first drawn at 3 months before the first child's birth, then at 3 months, 9 months and 36 months after childbirth. Each father's individual love trajectory is shown in Fig. 1. Data are grouped into three categories (low, medium, high positivity) based on a father's total number of positive experiences that occurred between marriage and 3 months before birth of their first child, measured by a 45-item checklist of positive events (e.g., job promotion, birth of family member; or reverse coded ones such as job loss, death of family member, etc.). The categories were created with equal number of subjects in each group. Note that the main purpose of this categorical binning is demonstrative, and including positivity as a continuous predictor would retain more information on the original measure. The visualization in Fig. 1 provides multifaceted observation of overall trends, group trends, and how widely individual trajectories vary.

Firstly, visualizing separate lines for each individual helps identify overall group variability in initial starting values, represented by the distance between lines at 3 months before the child was born. In Fig. 1 we observe lower levels of marital love for low positivity, as opposed to medium and high positivity. Secondly, observing the differing angles for each line helps identify overall group variability in growth trajectory (slope). Since we do not observe an overall common trend in slope, this indicates variability - and possible group differences - in growth. Moreover, the low positivity group exhibits the steepest decline in marital love over time, suggesting that fathers in this group have more negative love trajectories, compared to higher positivity groups, which appear to maintain more constant values of love. These preliminary visual observations help guide our attention to specific statistical follow-up measures.

### Specification of the linear growth curve model

A typical longitudinal data set contains  $T$  measurements at occasions  $t$  ( $t = 1, \dots, T$ ) from an individual  $i$ , where  $i = (1, \dots, N)$ , with  $N$  indicating the total number of people in the sample. The exact measurement time points can vary and need not be fixed across people; however,



**Fig. 1** Individual trajectories of father’s love scores across measurement occasions; group corresponds to the fathers’ level of positivity (1 = low, 2 = medium, or 3 = high number of positive life events post-marriage at first measure)

this tutorial considers fixed time points across people. The number of measurement occasions ( $T$ ) can also vary among participants, as the model flexibly handles various forms of missingness. The simplest assumption behind missingness is that it is unrelated to the modeled phenomenon, that is “missing completely at random” (MCAR, see in Little, 1995). If this is reasonable to assume, R users need no further action, since missing values are automatically coded as “NA” (“not available”), and will be automatically dealt with in proposed Bayesian engine or can be completely eliminated from the data (see in discussion on long format later with corresponding computer script). If MCAR does not apply, the missingness mechanism should also be included in the model.

The measure of love by father  $i$  at measurement occasion  $t$  is denoted by  $Y_{i,t}$ . In a simple growth curve model, we can express the change within-person over measurement occasions in terms of intercept (initial level) and slope (rate of change) parameters. That is, we are fitting a straight line to each father’s 4 measurements, with  $x$ -axis (independent variable) being the time, and  $y$ -axis (dependent variable) being the love measure. We can specify this GCM as follows:

$$Y_{i,t} \sim N(\beta_{i,1} + \beta_{i,2}T_t, \sigma_{e_{Level1}}^2) \quad (1)$$

$$\beta_{i,1} \sim N(\mu_{\beta_1}, \sigma_{e_{\beta_1}}^2) \quad (2)$$

$$\beta_{i,2} \sim N(\mu_{\beta_2}, \sigma_{e_{\beta_2}}^2). \quad (3)$$

In all three lines above,  $N$  stands for the normal (Gaussian) distribution. We specify the normal distribution in terms of mean and variance parameters. The tilde ( $\sim$ ) symbolizes “distributed as”, indicating that the parameters on the left hand side are assumed to follow the normal distributional form.

Equation 1 captures the effect of time at the person level, therefore it is often referred to as the level-1 equation. This specifies the likelihood function. In Eq. 1, the mean of our observed data  $Y_{i,t}$  is a function of a person  $i$ ’s intercept parameter  $\beta_{1,i}$  and the product between person  $i$ ’s slope parameter and the measurement occasion  $T_t$  at  $t$ , this way providing the conditional distribution of  $Y_{i,t}$  given  $\beta_{1,i}$  and  $\beta_{2,i}$ . The distributional shape of Eq. 1 is chosen to be Gaussian, with the time-specific residuals having variance  $\sigma_{e_{Level1}}^2$ . This allows for there to be error relative to the predicted person-specific change, with larger deviations becoming exponentially less likely. Note that the  $\sigma_{e_{Level1}}^2$  term could be modified to account for autocorrelation in the residual variation by adding an extra parameter to directly model this autocorrelation, and account for the time-dependency in the mean and variance structure captured by this parameter. Interested readers can consult the Git repository mentioned above for a worked out example.<sup>1</sup>

In contrast to our level-1 equation, Eqs. 2 and 3 are level-2, or population- (group) level equations, which capture between-person variability in initial levels (intercepts) and rates of change (slopes). In Eq. 2 parameter  $\mu_{\beta_1}$  is called the population mean or level-2 mean, and is a group parameter shared across participants. The variance term  $\sigma_{e_{\beta_1}}^2$  is the level-2 variation of the intercepts, representing the magnitude of the individual differences in initial values. Equation 3 describes the population level distribution of the slope parameters  $\beta_{i,2}$ : it has the shape of the normal distribution with  $\mu_{\beta_2}$  capturing the population mean slope and  $\sigma_{e_{\beta_2}}^2$  representing the individual differences in rates of change.

<sup>1</sup>See RAnalysisACerror.R file.

Later we will explicitly model the covariation between intercept and slope parameters. However, it is important to note that if we kept independent univariate priors on  $\beta_{i,1}$  and  $\beta_{i,2}$  they could still co-vary in the posterior distribution.

### Properties

As opposed to fitting a separate regression equation for each person's data, an important aspect of hierarchically modeling intercepts and slopes is that the level-2 or hyper-prior distributions pool information across subjects. In other words, a person's intercept and slope estimates are drawn from both individual- and population- (group) level information. This partial pooling is not particular to the Bayesian framework, nor does it happen only in the GCM: it is a general characteristic of hierarchical/multilevel modeling. To clarify, a completely pooled estimate would mean estimating only one intercept parameter and one slope parameter, which are the same for every person. A completely unpooled estimate would mean fitting a regression line separately for each individual's data over time. The person-level and population-level estimates in the GCM compromise between these two extremes: person-specific estimates are pooled towards group means, causing "shrinkage" towards the more general, population-level trend. In other words, the person - and population-level estimates share information: each individual's data contribute to the population mean, which in turn informs the person-specific terms.

Shrinkage of person-level intercept and slope estimates is a desirable attribute of multilevel modeling. Often within-person measurements are noisy, and we have fewer observations from some individuals than from others. Borrowing information from the whole group (from all data available) helps reduce the random sampling noise in person-specific estimates. Note that although the population trend constrains person-level estimates, person-level estimates are still pulled towards the individual's data. For the GCM in particular, it is interesting to highlight that even if we had one (or a few) participant(s) with only one observation, we could still estimate their person-specific intercept and person-specific slope parameters. Moreover in the Bayesian GCM these person-specific intercept and slope parameters have posterior distributions that quantify uncertainty around the estimates. Also, it turns out that Bayesian hierarchical modeling can provide relatively low uncertainty in these types of posterior estimates (for a worked example see Kruschke, 2015, Chapter 17).

Finally, we would like to emphasize the advantages of implementing these hierarchical models in the Bayesian framework in terms of accuracy of the estimates. Consider estimating the variation in person-specific intercept or slope parameters; that is in population level variances representing

individual differences. Often the variance components have a lot of uncertainty due to the fact that they capture variation in latent, person-specific constructs, which are themselves estimated with uncertainty. In Bayesian parameter estimation we integrate over the uncertainty in all the parameter values, meaning that the posterior uncertainty of the latent, person-specific constructs influences the posterior uncertainty in the population level constructs and vice versa.

### Specification of the linear GCM with a grouping variable

To examine individual variation in initial levels and rates of change, Eqs. 2 and 3 can be extended in several ways. In our application to the marital love study, we add a grouping factor based on fathers' levels of positivity. As stated in the introduction, we test whether fathers' baseline marital positivity - that is, the positivity fathers experienced in marriage, up until 3 months before the birth of their first child - can explain how their feelings of love towards their partner change post birth. Fathers were grouped into low, medium, and high positivity categories, based on self-reported numbers of positive events they had experienced at marriage to date (these reports were provided during the first measurement occasion, -3 months). We use fathers who scored medium positivity as our baseline group, and estimate their level-2 intercept and slope parameters. We model low and high positivity groups in terms of their deviations from medium positivity baseline. Compared to Eqs. 2 and 3, the GCM with grouping factors extensions (Eqs. 4 and 5 below) has additional components for interpretation, namely, group parameters with categorical, comparative estimates.

Whether a person belongs to a low or high positivity group will be coded by two dichotomous (dummy coded) 0-1 variables:  $X_{i,1}$  has value 1 for person  $i$ , if that person belongs to the low positivity group, while  $X_{i,2}$  has value 1 for person  $i$ , if that person belongs to the high positivity group. Persons belonging to the medium positivity group will have 0-s for both  $X_{i,1}$  and  $X_{i,2}$ . These  $X$ -s represent individual level, time-invariant predictors. Our new level-2 equations extend Eqs. 2 and 3 with these systematic group level variations as follows:

$$\beta_{i,1} \sim N(\mu_{\text{MedPInt}} + \beta_{\text{lowPInt}}X_{i,1} + \beta_{\text{highPInt}}X_{i,2}, \sigma_{e\beta_1}^2) \quad (4)$$

$$\beta_{i,2} \sim N(\mu_{\text{MedPSlope}} + \beta_{\text{lowPSlope}}X_{i,1} + \beta_{\text{highPSlope}}X_{i,2}, \sigma_{e\beta_2}^2). \quad (5)$$

As in Eqs. 2 and 3, intercept ( $\beta_{i,1}$ ) and slope ( $\beta_{i,2}$ ) parameters are person-specific, and therefore account for individual differences within groups. Parameters  $\mu_{\text{MedPInt}}$  and

$\mu_{\text{MedPSlope}}$  capture baseline intercept (initial value) and slope (rate of change) values for the medium level positivity group in our example. Regression coefficient  $\beta_{\text{lowPInt}}$  represents systematic deviations from baseline initial values (intercept for medium positivity group) in the low positivity group, while  $\beta_{\text{highPInt}}$  captures these for the high positivity group. Parameter  $\beta_{\text{lowPSlope}}$  represents deviations from the baseline rate of change (slope for the medium positivity group) in the low positivity group, while  $\beta_{\text{highPSlope}}$  captures these for the high positivity group. The likelihood

$$\begin{bmatrix} \beta_{i,1} \\ \beta_{i,2} \end{bmatrix} \sim N_2 \left( \begin{bmatrix} \mu_{\text{MedPInt}} + \beta_{\text{lowPInt}} X_{i,1} + \beta_{\text{highPInt}} X_{i,2} \\ \mu_{\text{MedPSlope}} + \beta_{\text{lowPSlope}} X_{i,1} + \beta_{\text{highPSlope}} X_{i,2} \end{bmatrix}, \begin{bmatrix} \sigma_{e\beta_1}^2 & \sigma_{e\beta_{12}} \\ \sigma_{e\beta_{21}} & \sigma_{e\beta_2}^2 \end{bmatrix} \right). \quad (6)$$

This mean vector of the bivariate distribution in Eq. 6 is a function of regression coefficients and predictors just like in Eqs. 4 and 5. Variation around the bivariate mean is expressed in terms of a covariance matrix. The elements of this matrix are  $\sigma_{e\beta_{12}}$ , which expresses covariation between person-specific intercepts and slopes, and  $\sigma_{e\beta_1}^2$  and  $\sigma_{e\beta_2}^2$ , which represent the variances of these terms. Dividing the covariance with the product of the standard deviation gives us the population-level correlation between intercepts and slopes.

### Prior specification

Now we specify prior distributions for all model parameters: normal distributions with mean zero and reasonably high variation for the group means for intercept and slope. The specification below is interpreted as setting a diffuse prior distribution on a wide range of possible values the parameter can take. Since we know that marital love was measured on a 9-point scale with 10 items, we use this knowledge to make sure we make the prior wide enough to fully cover the plausible range of the data. This corresponds to minimally informative priors on the baseline values (medium positivity group) and on the group-specific regression terms. The priors could be made even more diffuse, but this would negligibly differ from the chosen prior in its impact on the posterior. We specified the following normal priors, parameterized in terms of mean and variances:

$$\begin{aligned} \mu_{\text{MedPInt}} &\sim N(0, 100) \\ \mu_{\text{MedPSlope}} &\sim N(0, 100) \\ \beta_{\text{lowPInt}} &\sim N(0, 100) \\ \beta_{\text{highPInt}} &\sim N(0, 100) \\ \beta_{\text{lowPSlope}} &\sim N(0, 100) \\ \beta_{\text{highPSlope}} &\sim N(0, 100). \end{aligned}$$

specification, that is Eq. 1, is not repeated here as it remains the same.

In Eqs. 4 and 5 we specified level-2 distributions on intercepts and slopes univariately. However, traditionally these terms are allowed to co-vary. To have a more complete correspondence with the original GCM models we can also formulate the distribution of the person-specific intercepts and slopes bivariately. That is to say that we set a bivariate normal population (Level-2) hyperprior distribution on these parameters:

We set standard non-informative uniform distributions over a set of possible values for the error term, the standard deviations of intercept and slope, and on the correlation of these two terms (see more information on this prior choice in Barnard et al., 2000):

$$\begin{aligned} \sigma_{e_{\text{Level}2}} &\sim \text{unif}(0, 100) \\ \sigma_{e\beta_1} &\sim \text{unif}(0, 100) \\ \sigma_{e\beta_2} &\sim \text{unif}(0, 100) \\ \rho_{e\beta_{12}} &\sim \text{unif}(-1, 1). \end{aligned}$$

To get the covariance matrix, we can simply calculate the covariance from the standard deviations and correlation:  $\sigma_{e\beta_{12}} = \sigma_{e\beta_{21}} = \rho_{e\beta_{12}} \sigma_{e\beta_1} \sigma_{e\beta_2}$ . Note that in our current bivariate GCM specification, the correlation of intercept and slope parameter also has a posterior distribution and its likely range can be easily evaluated.

## Application: babies, fathers and love

### Study aims

To reiterate, the original purpose of this marital study was to assess aspects of marital change across the transition to parenthood. A longitudinal design was chosen to assess how trajectories for marital quality, measured from the last trimester of pregnancy through three years postpartum, varied in form and rates of change across individual fathers. In our application, we use a sample dataset that is a subset of the original dataset, consisting of fathers' measures of feelings of love and positivity scores.

## Subjects

Subjects in this dataset are 108 fathers. At time of enrollment, all were in intact marriages and expecting their first child. All families were Caucasian and of middle- and working-class socioeconomic status. For our purposes, we assess a subset of the original sample, containing only those fathers who had provided grouping measures for their pre-childbirth level of marital positivity.

### Love factor of marital quality

To identify patterns of marital change, Belsky and Rovine (1990) measured four aspects of the marital relationship: love, conflict, ambivalence, and maintenance. In the current study our sample dataset includes fathers' love measures only. The love scores are self-reports on the Braiker and Kelley (1979) love subscale, from fathers at -3 months, 3 months, 9 months, and 36 months relative to when they had their first child. This love subscale is a 10-item scale of intimate relations, assessing attitudes and beliefs about the parties in the relationship. Questions such as "to what extent do you have a sense of belonging with your partner?" (love) are answered on a 9-point scale (from very little or not at all, to very much or extremely). Internal consistencies of these scales across the four waves ranged from .61 to .92 for fathers and wives.

Positivity scores were constructed based on a 45-item life events checklist. This tool measured whether or not each listed event had taken place since onset of marriage, and measured the effect of the experienced events on a 7-point scale (low point = very negative effect, midpoint = no effect, high point = very positive effect). All fathers completed the checklist at measurement occasion 1, and ratings for positive effect responses were summed to create each individual's total positivity score.

## Step-by-step guide to fitting a Bayesian growth curve model in R

Here we provide step-by-step guidelines and computer script to fit GCM in the Bayesian framework by using R (with RStudio, RStudio Team, 2015) and JAGS (see e.g., in Plummer et al., 2003), which are open source statistical software packages. The most frequently used programs for fitting GCMs in the structural equation modeling framework include LISREL, Amos, and Mplus, whereas fitting GCMs as linear mixed models is more commonly done in SAS or SPSS. Some of these software packages have a Bayesian module in which GCM can be fitted, however all these programs include license fees.

Fitting models in JAGS provides more flexibility for custom extensions, including various prior specifications. As an alternative to using R to communicate with JAGS, MATLAB can also be used to formulate the Bayesian GCM via the Trinity package presented in this issue. See more details on the currently available programs in the Software options section later. Note that the Bayesian formulation of the GCM models applies across different statistical computing languages.

The following section explains how to fit a linear Bayesian GCM. We provide written explanation to guide the reader through five key modeling steps, with programming syntax that provides the necessary R code and output. The R output is always preceded by double hashtags. For execution in R Studio, the reader can copy our code from this paper directly, or access the corresponding R file or a .Rnw formatted version of this paper at the Git repository of the project.

**Step 0:** First install R, RStudio and JAGS. Then install the `rjags` package in RStudio so that R can work with JAGS.

---

```
# Step 0
# Installing the software
install.packages("rjags") # commented out when rjags is already installed
```

---

**Step 1:** Once the appropriate software is installed, we begin by reading the dataset into R and specifying which pertinent variables should be extracted for the analysis (note that in the script below it is assumed that the current working directory in R contains the data set). The code chunk below describes these steps: reading in the data, checking the data, counting the number of persons, extracting the 4 measurements

for each person (`data`), and defining the grouping variable (`grouping`, separated to `X1` for low positivity and `X2` for high positivity). Then we create the `time` vector by based on when the measurements were taken relative to the birth of the child (in month units). Finally, we create a list of all these variables in a format that can be read by JAGS, which is our generic Bayesian estimator engine for the analysis.

```

# Step 1
# Reading in the data
FathersLoveData <- read.csv("lovedata.csv")
# Visually checking the data
head(FathersLoveData)
##   Y1 Y2 Y3 Y4 Positivity X1 X2
## 1 79 83 70 75      3 0 1
## 2 80 81 85 63      3 0 1
## 3 81 73 80 78      3 0 1
## 4 78 80 76 79      1 1 0
## 5 78 83 79 62      1 1 0
## 6 79 67 77 77      2 0 0
# The first column(unnamed) is the person/row number:
# each participant's data corresponds to one row
# Y1-Y4 are the love scores for each person
# 'Positivity' is the positivity group for each person: 1: low, 2: medium, 3: high
# X1-X2 reflect positivity group via dummy coding:
# X1 takes value 1 for low positivity, otherwise 0
# X2 takes value 1 for high positivity, otherwise 0

# Checking the number of rows (i.e., 2nd dimension) of 'FathersLoveData' data
N <- dim(FathersLoveData)[1] # count number of rows to get number of subjects

# Creating a matrix with only love scores: Y1-Y4 (columns 1-4):
# We 'unlist' all N rows of selected columns 1:4 from the data set,
# then transform these values into numeric entries of a matrix
data <- as.matrix(FathersLoveData[,1:4])

# Creating a variable that indicates the number of time points
nrT <- 4

# Saving X1 and X2 as separate variables (same unlisting process as performed above)
grouping <- as.matrix(FathersLoveData[,6:7])
X1 <- grouping[,1] # 1 when person had low positivity before child
X2 <- grouping[,2] # 1 when person had high positivity before child

# Creating a time vector for the measurement waves
time <- c(-3, 3, 9, 36) # time vector (T) based on the time values of our measures
# Now we have all the data needed to be passed to JAGS

# Creating a list of all the variables that we created above
jagsData <- list("Y"=data, "X1"=X1, "X2"=X2, "N"=N, "nrT"=nrT, "time"=time)

```

**Step 2:** Next we write out the GCM in JAGS language, following the model specifications in Eqs. 1 and 6, and their corresponding prior specifications. In this code chunk, we use loop functions to handle the nesting structure in our data (i.e., multiple people in the population, multiple measures per person). First we create a population-level loop function over our multiple participants, then within this loop we

create an individual-level loop function over the multiple observations of an individual. At the center of these nested loops, we define the likelihood function (under the line `The likelihood`), which describes the assumed data generating distribution, as shown in Eq. 1. The shape of the likelihood is chosen to be normal. Due to programming requirements in JAGS, we specify the normal distribution



in terms of mean and precision, instead of mean and variance. Precision is defined as the reciprocal of the variance, therefore it is a simple one-to-one transformation: precision = 1/variance. Every time we specify a normal distribution, denoted with `dnorm` in JAGS, we will plug in the reciprocal of the variance as the second parameter. This is a technical matter and does not impact specification of the GCM.

Next, we close our inner, individual-level loop over observations and specify our person-specific hyperprior distributions, which correspond to the two person-specific parameters, the person intercept and the person slope (`beta[i,1:2]`). We call this a hyperprior distribution because it sets priors on hyperparameters, that is on

parameters of the population distributions of the  $\beta$ -s, as specified in Eq. 6. Technically, the prior distribution of the  $\beta$ -s is hyperparameterized, instead of its parameters being set to a fixed value, like we have seen in the Equations specifying priors above. We chose this population distribution to be a bivariate (or more generally, multivariate) normal distribution with a mean vector and a precision matrix, as shown in Eq. 6. The precision matrix is simply the matrix inverse of our covariance matrix in Eq. 6. As with `dnorm`, the JAGS language expects a precision matrix as an input for the multivariate normal distribution (denoted `dmnorm` in the syntax), and requires a one-to-one transformation from precision to variance.

```
# Step 2
# Specifying the growth curve model
LinearGrowthCurve = cat("
model {

# Loop over participants
for (i in 1:N) {
  # Loop over measurement occasions
  for (t in 1:nrT) {
    # Specifying likelihood function, corresponding to Equation 1:
    Y[i,t] ~ dnorm(betas[i,1]+betas[i,2]*time[t], 1/pow(sdLevel1Error,2))
  }

# Specifying the level-2 bivariate distribution of intercepts and slopes (Eq. 6)
betas[i,1:2] ~ dmnorm(Level2MeanVector[i,1:2], interpersonPrecisionMatrix[1:2,1:2])

# The mean of the intercept is modeled as a function of positivity group membership
Level2MeanVector[i,1] <- MedPInt + betaLowPInt*X1[i] + betaHighPInt*X2[i]
Level2MeanVector[i,2] <- MedPSlope + betaLowPSlope*X1[i] + betaHighPSlope*X2[i]
}

# Specifying prior distributions
MedPInt ~ dnorm(0,0.01)
MedPSlope ~ dnorm(0,0.01)
betaLowPInt ~ dnorm(0,0.01)
betaHighPInt ~ dnorm(0,0.01)
betaLowPSlope ~ dnorm(0,0.01)
betaHighPSlope ~ dnorm(0,0.01)
sdLevel1Error ~ dunif(0,100)
sdIntercept ~ dunif(0,100)
sdSlope ~ dunif(0,100)
corrIntSlope ~ dunif(-1,1)

# Transforming model parameters
# 1. Defining the elements of the level-2 covariance matrix
interpersonCovMatrix[1,1] <- sdIntercept * sdIntercept
interpersonCovMatrix[2,2] <- sdSlope * sdSlope
interpersonCovMatrix[1,2] <- corrIntSlope * sdIntercept* sdSlope
```

```

interpersonCovMatrix[2,1] <- interpersonCovMatrix[1,2]
# 2. Taking the inverse of the covariance to get the precision matrix
interpersonPrecisionMatrix <- inverse(interpersonCovMatrix)
# 3. Creating positivity variables representing:
# 3.1. Low positivity intercept
LowPInt <- MedPInt + betaLowPInt
# 3.2. High positivity intercept
HighPInt <- MedPInt + betaHighPInt
# 3.3. Low positivity slope
LowPSlope <- MedPSlope + betaLowPSlope
# 3.4. High positivity slope
HighPSlope <- MedPSlope + betaHighPSlope
# 4. Contrast terms between high-low, medium-low, high-medium intercepts and slopes
HighLowPInt <- betaHighPInt - betaLowPInt
MedLowPInt <- - betaLowPInt
HighMedPInt <- betaHighPInt
HighLowPSlope <- betaHighPSlope - betaLowPSlope
MedLowPSlope <- - betaLowPSlope
HighMedPSlope <- betaHighPSlope

}
",file = "GCM.txt")

```

Next, beneath the line `Specifying priors and transforming variables` we specify our prior distributions (non-hyperpriors) on parameters in line with the equations that described the priors. Lastly we create some new variables for planned comparison of our groups, (e.g., `LowPInt`; these variables all have the `<-` specification), which are one-to-one transformations of previously defined variables corresponding to our grouping levels. Inclusion of variable transformation is optional and depends on unique research questions; in our case it is interesting to add these contrast terms between the groups. The variable `HighLowPInt` for example represents the posterior probability distribution of the difference between the intercepts in the high and in the low positivity groups. The last line of code writes our model variable into file named `GCM.txt`.

**Step 3:** Next we estimate our Bayesian model parameters through sampling from their posterior distribution via MCMC algorithms, implemented in JAGS. To illustrate this process, we outline some important settings and specifications for the Bayesian estimation algorithm.

*Monitoring parameters.* First we create a vector called `parameters` in which we collect the names of all parameters that we want to examine. JAGS will save the posterior samples only for the parameters included in this vector. We

will first check whether the posterior sample chains have converged to the same area; more details on how to check the “convergence” will be provided below. The results are considered stable and reliable only after convergence criteria are met. Recall that we sometimes create parameters inside the JAGS code by simply transforming some other parameter(s) (i.e., computing the variance by squaring the standard deviation); these transformed parameters are necessary to include in the list if we want to draw inference about their likely values.

*Specifying sampler settings.* After collecting parameters, the remaining lines of Step 3 concern specifications for handling JAGS’s sampling algorithm. Since Bayesian posteriors are described through representative samples, we focus on how to handle our chosen algorithm to ensure that samples are accurate representations. Sampling is carried out in several *iterations*, typically in the range of thousands. In each iteration a new sample is drawn from the conditional posterior for each parameter (or a vector of parameters), and this new sample is conditional on the previously drawn value of all other parameters and the data. First, the sampling algorithm has to be adapted for the model and data: for this purpose 2000 adaptation iterations are typically sufficient. Second, while JAGS generates random initial values

for sampling, it is good practice to restart the sampling algorithm several times: that is to say several chains should be run (e.g., we recommend 6, although this number is somewhat arbitrary, and running only 3–4 chains can also be a good option when the burnin-in and adaptation iterations take a long time). The first iterations from our chains should always be discarded, as they are most likely influenced by the starting value of the chain and not yet converged to the target posterior area. This can be done by setting the burnin value to for example 1000: these first 1000 samples are drawn, but not included in the final posterior samples.

Within a chain the sampled parameter values might have high autocorrelation. We need a reasonable number of samples that are free from autocorrelation (as these contain more information, see below), therefore when high autocorrelation occurs, we must run longer chains. In complex models with many person-specific parameters long chains may exceed computer memory for storing and processing (calculating posterior summaries), an issue that may be

addressed by thinning. By thinning the chains we only save every  $x^{\text{th}}$  sample (e.g., in our example, we save every fifth sample), this way decreasing autocorrelation among consecutive samples, while also reducing the size of the chain resulting in less demand in terms of memory requirements. We choose to use thinning of 5. Link and Eaton (2012) showed that thinned chains have somewhat reduced information, thus thinning is only recommend if computer memory is an issue.

We focus on reducing autocorrelation in sample chains because lower autocorrelation means more representative random sampling – “quality” samples are barely correlated. We will later quantify the quality of our samples in terms of effective sample size (ESS). In the `postSamples` line we specify how many posterior samples we want for each parameter, our chosen value of 30000 (5000 for each chain) suffices for most models. The final line calculates how many number of iterations JAGS will need, as a function of number of chains and thinning, to achieve the required `postSamples` sample size.

```
# Step 3
# Monitoring parameters
parameters <- c("MedPSlope", "betaLowPInt",
               "betaHighPInt", "betaLowPSlope",
               "betaHighPSlope", "MedPInt",
               "sdIntercept", "sdSlope",
               "corrIntSlope", "sdLevel1Error", "betas",
               "LowPInt", "HighPInt", "LowPSlope", "HighPSlope",
               "HighLowPInt", "HighMedPInt", "MedLowPInt",
               "HighLowPSlope", "HighMedPSlope", "MedLowPSlope")

# Specifying sampler settings
adaptation <- 2000 # Number of steps to "tune" the samplers
chains <- 6 # Re-start the exploration "chains" number of times
burnin <- 1000 # Number of steps to get rid of the influence of initial values
thinning <- 5
# Defining the number of samples drawn from the posterior
postSamples <- 30000
# Computing the number of posterior samples needed per chain for JAGS
nrOfIter <- ceiling((postSamples*thinning)/chains)
```

**Step 4:** The package `rjags` (connecting JAGS with R) must be loaded into the R environment, as shown in the first R command line of the following code chunk. The next line uses an `rjags` function to compile a JAGS model object and adapt its samplers, it is named as `jagsModel` in our example. Then sampling is done in

two phases: a burnin phase (see `update` line, these first iterations are discarded), and a posterior sampling phase. Posterior samples are retained from this latter phase only, and are saved in a variable named `codaSamples` (note that `seed` is set to 5 here only to make this example replicable).

```

# Step 4
# Loading the rjags package
library(rjags)
# Creating JAGS model object
jagsModel<-jags.model("GCM.txt",data=jagsData,n.chains=chains,n.adapt=adaptation,
inits=fixedinits)
## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
## Observed stochastic nodes: 432
## Unobserved stochastic nodes: 118
## Total graph size: 5466
##
## Initializing model
# Running burn-in iterations
update(jagsModel,n.iter=burnin)
# Drawing posterior samples
codaSamples<-coda.samples(jagsModel,variable.names=parameters,n.iter=nrOfIter,
thin = thinning,seed=5)

```

**Step 5:** Once sampling is done, we can explore the posterior distributions of the model parameters of interest. We make use of function, `posteriorSumStats.R`, which can be found as an online resource on the Git project site. The script uses functions from the `coda` package from Plummer

et al. (2006) and from the utility script of Kruschke (2015). We note here that the `coda` package loads automatically with `rjags` and has built in summary and convergence checks functions that the reader might find useful, which calculates summary statistics from the posterior distributions.

```

# Step 5 source("posteriorSumStats.R")
# Part 1: Check convergence
resulttable <- summarizePost(codaSamples)
saveNonConverged <- resulttable[resulttable$RHAT>1.1,]
if (nrow(saveNonConverged) == 0){
print("Convergence criterion was met for every parameter.")
}else{
print("Not converged parameter(s):")
show(saveNonConverged)
}
## [1] "Convergence criterion was met for every parameter."
# Part 2: Display summary statistics for selected parameters
# with the summarizePost function. Type the name of the parameter of interest
# in the 'filter' argument, or use rules based on regular expressions
show(summarizePost(codaSamples, filters = c("^Med","^Low","^High","^sd", "^corr")))
##           mean    PSD PCI 2.50% PCI 97.50% 95% HDI_Low 95% HDI_High
## HighLowPInt    5.0164  1.8073   1.4358    8.5406   1.6039    8.6836
## HighLowPSlope  0.1707  0.0548   0.0627    0.2772   0.0622    0.2766
## HighMedPInt    2.0630  1.7687  -1.3876    5.5680  -1.4521    5.4967
## HighMedPSlope  0.0520  0.0539  -0.0531    0.1572  -0.0525    0.1573
## HighPInt       77.3477  1.3054  74.7688   79.9133  74.7455   79.8828

```

---

```

## HighPSlope      -0.0316  0.0393  -0.1083   0.0457  -0.1094   0.0445
## LowPInt         72.3313  1.2676  69.8572  74.8251  69.9044  74.8691
## LowPSlope      -0.2024  0.0386  -0.2777  -0.1266  -0.2776  -0.1265
## MedLowPInt     2.9533  1.7390  -0.5312   6.3260  -0.4635   6.3730
## MedLowPSlope   0.1188  0.0536   0.0135   0.2214   0.0129   0.2209
## MedPInt        75.2847  1.2210  72.8432  77.6665  72.9628  77.7849
## MedPSlope     -0.0836  0.0371  -0.1564  -0.0113  -0.1561  -0.0111
## corrIntSlope   0.2344  0.2498  -0.1935   0.8016  -0.2222   0.7593
## sdIntercept    6.7014  0.6123   5.5770   7.9637   5.5617   7.9427
## sdLevel1Error  5.7006  0.2817   5.1692   6.2731   5.1650   6.2671
## sdSlope        0.1222  0.0357   0.0451   0.1873   0.0478   0.1898
##
##               st -0.05  (-0.05,0.05)  lt 0.05  ESS  RHAT
## HighLowPInt    0.0028                0.0006  0.9966 26415 1.0002
## HighLowPSlope  0.0000                0.0139  0.9861 13328 1.0002
## HighMedPInt    0.1131                0.0122  0.8747 17256 1.0000
## HighMedPSlope  0.0290                0.4548  0.5163 6556  1.0000
## HighPInt       0.0000                0.0000  1.0000 26137 1.0000
## HighPSlope     0.3211                0.6587  0.0202 12185 1.0000
## LowPInt        0.0000                0.0000  1.0000 26020 1.0001
## LowPSlope      1.0000                0.0000  0.0000 13480 1.0002
## MedLowPInt     0.0432                0.0055  0.9513 17326 1.0000
## MedLowPSlope   0.0006                0.0997  0.8997 7174  1.0001
## MedPInt        0.0000                0.0000  1.0000 16170 1.0000
## MedPSlope      0.8194                0.1805  0.0001 5994  1.0000
## corrIntSlope   0.1153                0.1227  0.7619 1973  1.0040
## sdIntercept    0.0000                0.0000  1.0000 16761 1.0000
## sdLevel1Error  0.0000                0.0000  1.0000 5139  1.0010
## sdSlope        0.0000                0.0333  0.9667 1383  1.0039

```

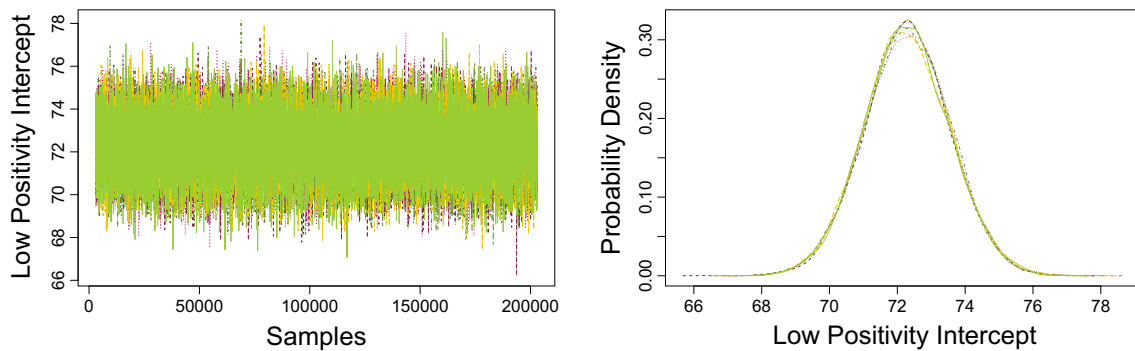
---

*Part 1.* The first R command of the Step 5 code chunk calls this function into the R environment. The subsequent lines check convergence. As mentioned above, for each parameter, we ran several chains with disperse starting values to explore the posterior distributions. For our results to be reliable, we must confirm that the chains converge to the same area, per parameter; this ensures that all chains are tuned to find similar likely values for the parameter.

We recommend graphical and numerical convergence checks of the posterior distribution. Figure 2 shows two graphical checks of the level-2 high positivity intercept parameter.<sup>2</sup> The plot on the left depicts the six sample chains: we can see that the chains overlap very well, indicating that they converged to the same area. The plot on the right shows the smoothed posterior probability densities for the same parameters, depicted with different colors for each chain. These densities also nicely overlap, supporting convergence.

<sup>2</sup>Computer scripts generating the figures are available in the Git repository of the project.

Aside from graphical checks, we use  $\hat{R}$  statistic (see, e.g., in Gelman et al., 2013) as a numerical indicator for convergence. The  $\hat{R}$  statistic is a ratio of the between and the within chain variances. If all chains converged to a region of representative samples, then the variance between the chains should be more or less the same as the mean variance within the chains (across iterations). Conventionally,  $\hat{R} < 1.1$  suggests that the chains for a certain parameter reached convergence. While this criterion is commonly set at 1.1, it can be changed to be more conservative (e.g., 1.05). In the code block above, there is a `if` statement to send a confirmation message if all parameters converge (as below, “Convergence criterion was met for every parameter.”), otherwise R will display the name and posterior statistics of the unconverged parameters. If this unconverged parameter table appears (an example is not shown here) and the  $\hat{R}$  values are around 1.2, you can often solve the problem by re-running the analysis with an increased required posterior sample size (`sizeofPost`).  $\hat{R}$  values above 2 most likely refer to serious misfit between model and data, or coding error. Results should only be interpreted when chains are converged for all parameters.



**Fig. 2** Graphical checks for convergence: traceplots (*left*) and smoothed posterior probability densities (*right*) of the six sample chains. The overlapping chains in both plots support convergence

*Part 2.* The second part of the code chunk calculates posterior statistics for the parameters of interest: intercept and slopes in the low, medium and high positivity groups, contrast terms between the groups, standard deviations and correlation between intercepts and slopes. A useful posterior point estimate for parameters of interest is the mean of the posterior distribution. The PSD column shows the posterior standard deviation, which quantifies the uncertainty around the point estimate (similarly to a standard error in the classical statistical framework). The 2.5% and 97.5% columns designate the two ends of the 95% posterior credibility interval (PCI), which is the center 95% of the posterior probability distribution of the parameter: parameters will fall in this interval with probability .95. Next to these are the low and high ends of the 95% highest probability density (HDI) interval: this interval designates the 95% range of values with the highest probability density. As seen in our summary table, the limits of PCI and HDI are almost identical, due to the fact that the posterior distribution is approximately symmetrical. However, for skewed distributions the equal tailed PCI might exclude values with high probability density in favor of values with low probability density.

The next three columns concern statistics related to the Region of Practical Equivalence (ROPE, defined later). The script `posteriorSumStats.R` automatically sets this region to be -0.05 and 0.05, but the limits of the ROPE can be tailored depending on the questions of interest.<sup>3</sup> For a specified ROPE we calculate the probability that a certain parameter is smaller than the lower limit of ROPE (column 7), falls within the specified ROPE interval (column 8), and is larger than the upper ROPE limit (column 9).

As referenced in our discussion of “quality” samples above, the ESS column of the summary table measures

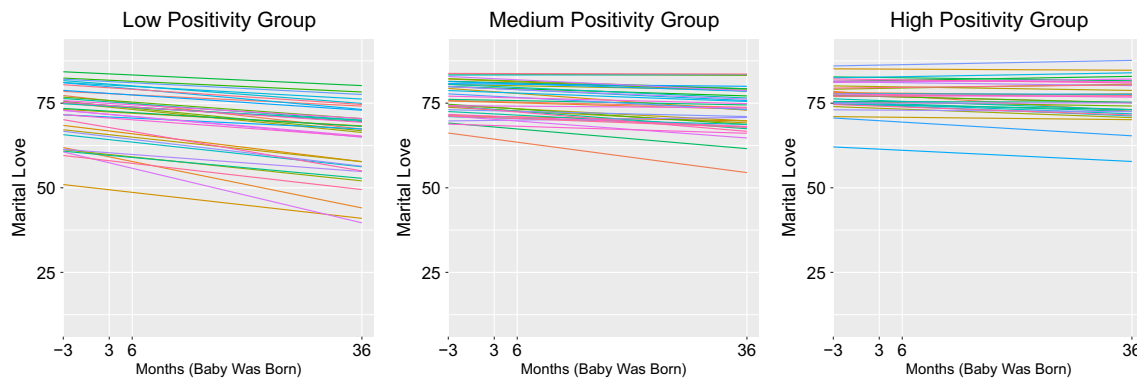
effective sample size: number of total posterior samples that do not correlate substantially. ESS counts our “quality” samples. As a general rule of thumb we should have a couple of thousands effective samples per parameter; Kruschke (2015, Section 7.5.2) recommends 10,000 for getting stable estimates on the limits of some selected HDI. As can be seen some of the reported parameters have somewhat less than 10,000 effective samples. For these cases we could simply draw more iterations to achieve a higher ESS, but we decided not to as we are not interested in their HDI limits (e.g., for population variance parameters). Lastly, the  $\hat{R}$  column in the summary table shows the  $\hat{R}$  statistics values discussed above in the context of convergence. Values very close to 1 suggest good convergence; most of our parameters have this value, and none has  $\hat{R}$  larger than 1.1.

## Results

### Graphical illustrations

The Bayesian GCM helps us articulate how men experience marital love during their transition to fatherhood. Per the summary table above, results of this study suggest that husbands’ experience of marital love across the transition to fatherhood is adequately explained by a linear Bayesian GCM, which accounts for within- and between-person dynamics. While not shown in this paper to conserve space, the above GCM analysis provides us with estimates of all person-specific ( $N = 106$ ) intercept and slope parameters. These person-specific estimates have probability distributions and the most probable range of values can be inferred. The user can print these estimates by adding the variable names (‘betas’) into the filter argument of the `summarizePost` function in Step 5, Part 2. Figure 3 shows the model predicted person-specific slopes based on the posterior mean estimates of the person-specific parameters.

<sup>3</sup>For this add argument: e.g., `ROPE=c(2, 4)`, for limits 2 and 4, when calling the function. This mean that you designated a region around 3, extending plus/minus 1 in each direction.



**Fig. 3** Person-specific trajectory estimates

Compared with the raw data trajectories in Fig. 1, the estimated lines in Fig. 3 reflect both person-specific (each line) and group-specific (each panel) patterns. The person-specific estimates shrink away from the raw observations in Fig. 1 and towards the estimated population trajectory, illustrated in Fig. 4 (see more on that below). By using multiple levels of information, these estimated trajectories shift in magnitude to their sample size (i.e., smaller samples shrink more, and benefit most from shrinkage) and exhibit better out-of-sample prediction.

Figure 4 gives a graphical illustration of the level-2 results. The thick lines illustrate the population slope for a given group, and the surrounding area shows the posterior uncertainty around the population slope estimate, based on the posterior samples. As in Fig. 3, trends governed by the time-invariant positivity grouping factor can be noticed in Fig. 4: low and high positivity groups visibly differ in their love trajectories. Although we can visually spot differences in these plots, we next examine whether there is enough supporting evidence to confirm there are meaningful differences in group trends.

### Numerical summaries

The Bayesian GCM yields a multi-parameterized solution, however here we extract only the most pertinent estimates for interpretation. As can be seen by comparing intercept values across positivity groups, the higher a father's positivity level, the higher his level of marital love, with intercept values quantifying the (linearly interpolated) levels of felt love at childbirth: high positivity intercept ( $M = 77.3$ , 95% HDI = (74.7, 79.9)), medium positivity intercept ( $M = 75.3$ , 95% HDI = (73.0, 77.8)), low positivity intercept ( $M = 72.3$ , 95% HDI = (69.9, 74.9)).

When it comes to slope estimates or differences among groups, it is useful to ask how likely it is that these variables would differ from 0, or be practically equivalent to 0. We can designate a Region of Practical Equivalence (or

ROPE, see more discussion in Kruschke, 2015, Section 12.1) around 0, which defines a range of parameter values that we consider to be practically equivalent to 0. The ROPE range is typically small, with limits that depend on the substantive meaning of the parameters. In other words, ROPE answers the question: what is a negligible amount of deviation from 0, with respect to the problem at hand? For example in the current analysis, we selected a ROPE with upper and lower limits  $-0.05$  and  $+0.05$  for the raw-scale regression coefficients that expressed the association between the outcome measure (ranged between 0 and 100) and the measurement time points (ranged between  $-3$  and 36). Multiplying the predictors with the very small values contained in the ROPE results in a negligible change on the scale of the outcome.

With a chosen ROPE, we can directly quantify (1) the percentage of the posterior probability distribution that falls within the Region of Practical Equivalence to 0 (column  $(-0.05 \ 0.05)$  in the R output), (2) the percentage of the posterior that is larger than the upper limit of the ROPE (column  $1 \leq 0.05$  in the R output), supporting positive values for the coefficient, and (3) the percentage of the posterior that is smaller than the lower limit of the ROPE (column  $\leq -0.05$  in the R output), supporting negative values for the coefficient. Note also that based on the ROPE and the 95% HDI, Kruschke (2015) proposes to set up a decision rule by checking whether the ROPE includes the 95% HDI, or whether the 95% HDI is completely below/above the ROPE, leading to conclusions such as that the 95% most credible parameter values are practically equal to or less/more than 0, respectively. If the ROPE and the 95% HDI partly overlap this decision rule remains undecided based on the current data (see more discussion in Kruschke, 2015, Section 12.1). To allow for this type of conclusion, we report the HDI alongside with the proportion of posterior mass inside or outside the ROPE.

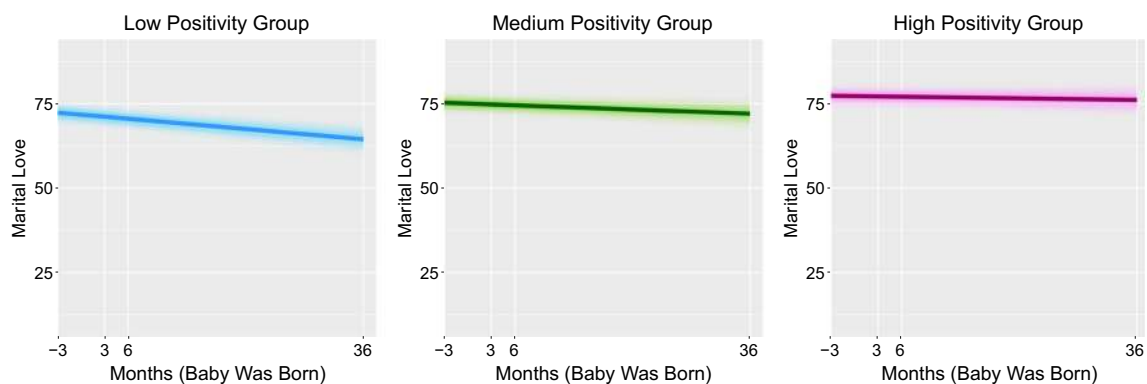
First, we explore the results on the group specific slope coefficients (rates of change in self-reported marital love).

The posterior mean for the low positivity group is  $-.20$  (95% HDI =  $(-0.28, -0.13)$ ), with the entire posterior probability distribution falling on the left (negative) side of the ROPE ( $st - 0.05 = 1$ ), which leads us to conclude that there is a negative slope for the low positivity group. Substantively, this suggests that fathers who reported a low number of positive life events before childbirth experienced decreasing feelings of marital love over the transition to parenthood. In contrast, fathers with high or medium pre-birth positivity did not exhibit remarkable upward or downward trajectories. The medium positivity group had a slight negative slope ( $M = -.08$ ), with a lot of uncertainty around the estimate: the 95% HDI ranges from  $-0.16$  to  $-0.01$ , which partly overlaps with the ROPE, and with 82% of the probability mass on the negative side of the ROPE ( $st - 0.05 = 0.82$ ). The high positivity group's trajectory is practically flat, with most likely values centered on zero:  $M = -.03$  (95% HDI =  $(-0.11, 0.04)$ ) and the probability that this slope is practically equivalent to 0 is 66% ( $(-0.05, 0.05) = 0.66$ ).

Specific to our research interests, we aim to assess whether there are statistically meaningful differences in the intercepts and slopes of the low, medium and large positivity groups. Recall that we specified contrast terms to compare the 3 groups' intercepts and slopes in the model fitting algorithm (Step 2 above): for example `HighLowPInt` represented the difference between the credible values of the intercept in the high and in the low positivity groups. Each contrast term has its own posterior probability distributions. In case of the `HighLowPInt`, the posterior distribution is based on subtracting the sampled value of the low positivity intercept from the high positivity intercept at every iteration. We specified 6 contrast terms, representing differences between groups, in terms of intercepts and slopes: `HighLowPInt`, `HighLowPSlope`, `HighMedPInt`, `HighMedPSlope`, `MedLowPInt`, `MedLowPSlope`. The posterior mean of the contrast term summarizes the magnitude of these group differences, and the 95% HDIs quantify the amount

of uncertainty around these group differences. Based on these results, the high and low positivity groups showed remarkable differences: (1) the high-positivity group, compared to low-positivity group, was higher in mean levels of felt love with 5.0 magnitude (difference in intercepts, `HighLowPInt`, 95% HDI =  $(1.6, 8.7)$ ,  $1t - 0.05 = 1\%$ ) and (2) the high-positivity group, compared to low positivity group, experienced slightly less decline ( $M = 0.17$ ) in felt love over time (difference in slopes, `HighLowPSlope`, 95% HDI =  $(0.06, 0.28)$ ,  $1t - 0.05 = 0.99$ ). These differences between low and high positivity groups can be spotted when looking at the left and right panels of Fig. 4. As for the rest of the contrast terms, the medium-positivity group, compared to the low-positivity group, experienced slightly more love ( $M = 3.0$ ) and less decline in felt love ( $M = 0.12$ ). However, there is considerable uncertainty around these estimates, reflected by the relatively wide 95% HDIs, partly overlapping with the ROPE (`MedLowPInt`, 95% HDI =  $(-0.5, 6.4)$ ; `MedLowPSlope`, 95% HDI =  $(0.01, 0.22)$ ). Finally, we did not find remarkable differences between high and medium positivity groups in terms of intercept and slope (`HighMedPInt`,  $M = 2.1$ , 95% HDI =  $(-1.5, 5.5)$ ; `HighMedPSlope`,  $M = 0.05$ , 95% HDI =  $(-0.05, 0.16)$ ). To conclude, overall we find interesting differences between high and low positivity groups; however the effect sizes are rather small.

With respect to the Level-2 covariance matrix defined in Eq. 6, results are summarized in terms of standard deviations and correlation in the R output. The person-specific intercept and slope terms showed a slight positive correlation (`corrIntsSlope`,  $M = 0.23$ ), but the 95% HDI for this estimate ranged from  $-0.22$  to  $0.76$ , indicating a lot of uncertainty in this estimate. Parameter `sdSlope` represents the individual differences in slopes, and had a posterior mean of  $0.12$  with 95% HDI ranging from  $0.05$  to  $0.19$ . The standard deviation parameter representing individual differences in intercepts (`sdIntercept`) had posterior mean  $6.7$ , with 95% HDI from  $5.6$  to  $7.9$ , which



**Fig. 4** Group differences. The *thick line* is based on the population values, and the surrounding area is based on all the posterior samples



corresponds to sizeable individual differences considering the scale of the outcome. Finally, the posterior estimate for the standard deviation on level-1 (`sdLevel1Error`) is 5.7 with 95% HDI from 5.2 to 6.3. The ROPE related probabilities, given the  $-0.05$  and  $0.05$  limits are also displayed in the R output for these three standard deviation parameters, however, they are less useful here as standard deviation parameters are constrained to be positive.

Next we assess how appropriate our complex model is for this data set by comparing the fit of the above GCM to a more complex and some simpler models.

### Model comparison via the Deviance Information Criterion

To assess the overall performance of our fitted GCM, we may evaluate its relative goodness of fit in comparison to other models. Our chosen fit statistic for this purpose is the Deviance Information Criterion (DIC, Plummer 2008). DIC is a Bayesian information criterion that quantifies the information in the fitted model by measuring how well the model reduces uncertainty of future predictions. Adding more parameters most often improves model fit, but overly complex models may risk overfitting to sample data; overfitted models yield poor out-of-sample prediction. In traditional measures of goodness-of-fit such as explained variance (e.g.,  $R^2$ ), increases in model fit do not penalize for the amount of parameters included in the model, regardless of complexity. DIC, on the other hand, simultaneously accounts for model complexity (number of parameters) and model fit, by penalizing based on the number of (effective) parameters. DIC is calculated based on the sum of the effective number of parameters and the posterior mean of the

**Table 1** Deviance Information Criterion values for five models fitted to the marital love data

Model type	DIC
Quadratic change with positivity grouping	2890
Linear change with positivity grouping	2856
Linear change with no positivity grouping	2855
No change with positivity grouping	2925
No change with no positivity grouping	4905

Note that these models all allow for person-specific intercepts. The Linear Change model, with positivity grouping, is the central model presented in this paper. Lower DIC values indicate better model performance in predicting future values

deviance, with deviance defined as  $-2$  times the log of likelihood function.

When comparing several models fitted to the same data, models with smaller DIC have less out-of-sample deviance, and thus will yield more accurate future predictions about populations similar to our sample. It should be noted that DIC has limitations. Since DIC is not based on  $n$  model probabilities, models cannot be compared in probabilistic terms, only in terms of their relative goodness of fit. Also, DIC is based on the assumption that the posterior distribution has a multivariate normal shape. Bayesian goodness of fit indices continue to be assessed in the literature (see, e.g., Gelman et al., 2014). Finally, we point out again that DIC can only evaluate the predictive performance of a model in comparison to others. Absolute model fit in the Bayesian framework is typically evaluated via posterior predictive checks (see, e.g., Kruschke, 2015, Section 17.5.1 and Gelman et al., 2013, Section 6.3).

DIC calculation in JAGS is easily implemented with the following lines:

```
# Calculating Deviance Information Criterion for the fitted model
DICSamples <- dic.samples(jagsModel, nrOfIter, thin = thinning, "pD")
show(DICSamples)
## Mean deviance: 2728
## penalty 128.6
## Penalized deviance: 2856
```

To assess performance of our fitted Bayesian GCM, we compare our model (denoted as “Linear change with positivity grouping”) to four other models: one more complex, and three simpler models. The more complex model allows change over time to take a quadratic curvature (“Quadratic change with positivity grouping” model). This model has an extra person specific regression coefficient ( $\beta_{i,3}$ ), which would add an extra dimension to Eq. 6. A worked example of this model can be found as an online supplement.<sup>4</sup>

<sup>4</sup>See `scriptForDICCalculations.R` file.

When it comes to simpler models, one of the simpler models we included does not add the grouping variable based on fathers’ initial level of positivity in marriage (“Linear change with no positivity grouping”), since not all the contrast terms capturing group differences were convincingly different from 0. Another simpler model (“No change with positivity grouping”) assumes no change in marital love levels over time, and participants in the three groups are simply described by their means across the four measurement occasions. In other words, this model has no slope parameter, but allows for individual differences in the

intercept. Finally, the simplest model “No change with no positivity grouping”, excludes the positivity grouping factor but allows for a random intercept for each person.

The resulting DIC values are displayed in Table 1. The models are ordered according to their level of complexity, starting with the most complex one on the top and ending with the most parsimonious one in the bottom. As mentioned, a lower DIC value indicates better model performance in predicting future values, therefore, based on DIC the model with linear change but no positivity grouping is the preferred model of this ensemble, with  $DIC = 2855$ . However, the difference between this simpler model and our featured model with positivity grouping is only 1 point, which is not interpreted as considerable difference in DICs (typically values larger than 5 indicate some difference and values above 10 are considerable, for more discussion see Lunn et al., 2012). There is considerable increase in DIC when the slope parameter is removed (simpler models), and adding a quadratic term does not substantially improve future predictive accuracy.

To summarize, the more parsimonious model without positivity grouping shows similar performance in predicting future values as our model. Our model is not considerably worse and might allow for exploring some substantively interesting aspects of how changes in felt love marital love are related to initial levels of positivity.

## Further considerations

### Different patterns of growth

An additional straightforward GCM extension is to model different shapes of the growth curve, beyond just a straight line. For example, as already discussed in the model comparison section above, polynomial growth may be measured in terms of a quadratic shape. With this extended model we capture individual differences not only in terms of initial values and rate of change, but also acceleration or deceleration. This extension requires adding one extra term to the mean structure in Eq. 1, namely  $\beta_{i,3}T_i^2$ , where  $\beta_{i,3}$  stands for the person-specific acceleration and  $T_i^2$  is the measurement time vector squared.

An alternative polynomial extension involves adding a cubic term, that is  $T_i^3$ . For that we need to add yet another term,  $\beta_{i,4}T_i^3$  to the mean structure in Eq. 1. Extending with quadratic and cubic terms yields more complex interpretations: our longitudinal growth process now contains higher-order features of rates of change, acceleration, and curvature, which all depend on (i.e., are multiplied by) the time index. Another complexity is that in our marital dataset we only have 4 time points per individual and extending up to cubic term would mean estimating as many person-specific parameters as we have data points per person. In

other words, a cubic GCM would be considered a saturated model for this dataset, according to the classical framework. We note here that the individual level parameters would be constrained by both hierarchical shrinkage and Bayesian hyperpriors, therefore while this model appears saturated it still leaves some space for generalizability in the Bayesian framework. Overall, for longitudinal psychology research, the most pertinent polynomial functions have curvilinear forms, typically up to quadratic order.

Besides polynomial linear models of increasing degree, the GCM framework offers a wide range of possibilities for modeling change over time. For example spline functions can be useful growth functions for processes which appear to follow a growth trajectory that dramatically changes form at a transition, or knot point. Spline functions hinge on a transition point in our time index, in which the function of growth completely changes. These models have useful interpretations for developmental studies of transitions between major life phases, such as from childhood to adolescence, or of transitions before and after a major life event. To specify a spline model requires two equations - one which describes the growth (linear or nonlinear) prior to transition, and one which describes growth (linear or nonlinear) after transition. More information can be found in McArdle et al. (2008).

Finally the most popular non-linear GCM is one that models exponential growth (see e.g., in Grimm et al., 2011). Interested researchers should also reference Ram and Grimm (2015) for a good summary of growth models include curvilinear, latent basis, exponential, sigmoid, sinusoid, and spline functions and their applications in psychological research. The exponential model, with either additive or multiplicative random coefficients, serves as a useful nonlinear approach for modeling developmental processes exhibiting curved growth towards an asymptote, and have been applied to studies including cognitive, learning, and language development, due to its straightforward application. Exponential curves have interpretable advantages for asymptotic growth, and these functions are able to model complex developmental patterns with few parameters. However, if a study's measurement occasions miss the asymptotic growth phase, we might mistakenly choose an exponential model over a logistic growth process. Depaoli and Boyajian (2014) provide an accessible summary of non-linear (and linear) models implemented in the Bayesian framework, and argues that GCMs fit via non-Bayesian estimation techniques can yield inaccurate parameter estimates under certain conditions. For careful consideration of nonlinear models, readers can consult for example Ghisletta et al. (2010) or Grimm and Ram (2009).

Most GCM extensions can be implemented in a straightforward manner into the JAGS script presented in this paper. Results for additional parameters can be interpreted in the same manner as demonstrated above: we can look at the

posterior of each parameter directly and make statements in probabilistic terms about the likely values of the parameters. It is crucial advantage of the Bayesian framework that once the researcher becomes familiar with the Bayesian model structure and computation of a basic model (like the example shown above), all further extension can be formulated in a straightforward and transparent manner.

### Long format

In the current application we use a data frame structured in person-by-observations format, which is sometimes called “wide” format. Typically, however, we prefer a less concise but more flexible representation called “long” format: that is all repeated observations are stacked under each other in a long vector. In this format the data frame includes two additional columns for hierarchical indexing and nesting purposes: a person-ID column, and a time column, in the metric of the measurement occasion (e.g., seconds, days, waves).<sup>5</sup> This formatting adds extra flexibility to the model. For example, it offers straightforward implementation for unbalanced, unstructured and unequally spaced data. Moreover, we can simply delete the missing observations (given the missing at random assumption is met) from the data vector and its corresponding index vectors, this way saving computation time.

### Software options

For researchers not familiar with the R environment, OpenBUGS’ point and click interface offers a user friendly alternative (see examples in Lunn et al. 2012). The current JAGS script can also be easily adapted to be run from BUGS. Another easy to use package designed specifically for fitting latent variable models (in the classical and in the Bayesian framework) is Mplus. Depaoli and Boyajian (2014) provide computer code examples for fitting Bayesian growth curve models in OpenBUGS and Mplus.

A more generic, flexible software is Stan, which was specifically developed for hierarchical modeling. As opposed to JAGS or BUGS, Stan uses a different sampling algorithm that may more efficiently estimate complex growth curve models with higher order time effects. Stan model specification involves additional specification for model parameters and data, but shares several common features with the JAGS syntax. Kruschke (2015) and McElreath (2016) provide description and annotated computer code (on the authors’ websites) for some linear and quadratic growth curve models, implemented both in Stan and JAGS. Finally,

the `rstanarm` package in R provides a user-friendly solution for specifying GLMMs (concise syntax, similar to `lme4`, see more in Gabry & Goodrich, 2016), while relying on Stan to estimate the model parameter.

### Computational considerations

To reiterate, the computational time demand is considerably higher for Bayesian estimation than MLE. Especially for complex GCMs, with correlated parameters, the MCMC algorithms implemented in JAGS take time to collect quality samples from highest density region, as discussed in regards to thinning earlier. When this computational burden becomes prohibitive, one should consider re-parameterizing the model for more efficient sampling of posteriors (see, e.g., in Browne et al., 2009; Gelfand et al., 1995).

### Complex issues in growth curve modeling

The simple GCM presented in this paper, and the extensions described above, cover a large part of the modeling needs in substantive applications. However, researchers may have more complex GCM questions that can only be addressed with highly versatile models. The Bayesian framework has shown great potential in handling the complex issues involved in fitting these models. For example, Depaoli (2013) shows that when fitting growth curve models with latent classes (growth mixture models) optimal parameter recovery can only be obtained in the Bayesian framework with informative priors. Lu and Zhang (2014) highlight the advantages of Bayesian modeling in handling non-ignorable missingness in growth mixture models. We expect that the number of sophisticated extensions to growth curve models will increase rapidly in the future.

### Discussion

In this study we argued that implementing Bayesian estimation procedures for growth curve modeling is acceptable in terms of ease via generic Bayesian inference engines such as JAGS. Although computation time is often higher in the Bayesian framework than with MLE, the benefits - namely, flexible extension and results with rich, intuitive probabilistic interpretations - are substantial. The Bayesian framework also allows for incorporating existing knowledge on the likely values of the growth curve model parameters, via the prior distribution when such information available.

Beyond a wide array of statistical advantages to modeling with posterior distributions, the Bayesian framework offers robust estimating approaches for fitting the GCM. Previous studies have shown this robustness by demonstrating: (1) that Bayesian GCMs with uninformed priors

<sup>5</sup>See our worked out example in long format in file `Ranalysislongformat.R` in the Git repository.

yield maximum a posteriori (MAP) estimates that are identical to maximum likelihood estimates; and (2) that the use of increasingly informative priors serve to reduce standard deviations (Bayesian standard errors) of Bayesian GCM estimates, as shown by Zhang et al. (2007). Zhang (2013) have also demonstrated the robust nature of Bayesian GCMs to handle nonnormal residual variance via the generalized error distribution without adding error to inference. Zhang et al. (2013) and Kruschke (2015) also demonstrate how the likelihood function for the growth curve model can be easily adapted to the unique features of a given dataset; for example to adapting the likelihood to a t-distribution to accommodate outliers (see e.g., in Kruschke, 2015, Section 16.2).

In our example of marital love, GCM parameter estimates facilitated assessment of the psychological process in question. By accounting for within- and between-person trajectories, as well as grouping factors, we find meaningful measures to describe our subjects' multifaceted emotional change process. We demonstrated how our proposed Bayesian analysis can be implemented in JAGS and how posterior distributions of the parameters of interest can be calculated and interpreted in straightforward, probabilistic terms.

We end with a final note on the advantage of using Bayesian estimation for GCM, as we enter a new era of longitudinal data analysis. Although traditional GCMs have captured incremental linear change processes, longitudinal applications and time scales are rapidly expanding, and researchers must consider more complex functional forms of growth. The Bayesian approach provides flexible tools to estimate parameters of complex change functions in the GCM framework.

**Acknowledgments** We would like to thank Mike Rovine for generously providing us the dataset on marital happiness.

The research reported in this paper was sponsored by grant #48192 from The John Templeton Foundation.

## References

- Barnard, J., McCulloch, R., & Meng, X.-L. (2000). Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica*, 10(4), 1281–1312.
- Belsky, J., & Rovine, M. (1990). Patterns of marital change across the transition to parenthood: Pregnancy to three years postpartum. *Journal of Marriage and the Family*, 5–19.
- Boker, S.M., & Wenger, M.J. (2012). *Data analytic techniques for dynamical systems*. Psychology Press.
- Bollen, K.A., & Curran, P.J. (2006). *Latent curve models: A structural equation perspective* (Vol. 467). John Wiley & Sons.
- Braiker, H.B., & Kelley, H.H. (1979). Conflict in the development of close relationships. *Social Exchange in Developing Relationships*, 135–168.
- Brown, S., & Heathcote, A. (2003). Averaging learning curves across and within participants. *Behavior Research Methods, Instruments, & Computers*, 35(1), 11–21.
- Browne, W.J., Steele, F., Gollalzadeh, M., & Green, M.J. (2009). The use of simple reparameterizations to improve the efficiency of markov chain monte carlo estimation for multilevel models with applications to discrete time survival models. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 172(3), 579–598.
- Curran, P.J., & Bollen, K.A. (2001). The best of both worlds. Combining autoregressive and latent curve models.
- Depaoli, S. (2013). Mixture class recovery in gmm under varying degrees of class separation: Frequentist versus bayesian estimation. *Psychological Methods*, 18(2), 186.
- Depaoli, S., & Boyajian, J. (2014). Linear and nonlinear growth models: Describing a bayesian perspective. *Journal of Consulting and Clinical Psychology*, 82(5), 784.
- Diggle, P., Heagerty, P., Liang, K.-Y., & Zeger, S. (2002). *Analysis of longitudinal data*. Oxford University Press.
- Ferrer, E., & McArdle, J.J. (2004). An experimental analysis of dynamic hypotheses about cognitive abilities and achievement from childhood to early adulthood. *Developmental Psychology*, 40(6), 935.
- Gabry, J., & Goodrich, B. (2016). rstanarm: Bayesian applied regression modeling via stan. R package version, 2, 0–3.
- Gelfand, A.E., Sahu, S.K., & Carlin, B.P. (1995). Efficient parametrizations for normal linear mixed models. *Biometrika*, 82(3), 479–488.
- Gelman, A., & Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press.
- Gelman, A., Carlin, J.B., Stern, H.S., & Rubin, D.B. (2013). *Bayesian data analysis*, third ed. Boca Raton, FL.: Chapman & Hall/CRC.
- Gelman, A., Hwang, J., & Vehtari, A. (2014). Understanding predictive information criteria for bayesian models. *Statistics and Computing*, 24(6), 997–1016.
- Ghisletta, P., Kennedy, K.M., Rodrigue, K.M., Lindenberger, U., & Raz, N. (2010). Adult age differences and the role of cognitive resources in perceptual–motor skill acquisition: Application of a multilevel negative exponential model. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, 65(2), 163–173.
- Grimm, K.J., & Ram, N. (2009). Nonlinear growth models in m plus and sas. *Structural Equation Modeling*, 16(4), 676–701.
- Grimm, K.J., Ram, N., & Hamagami, F. (2011). Nonlinear growth curves in developmental research. *Child Development*, 82(5), 1357–1371.
- Hamaker, E.L. (2012). Why researchers should think “within-person”: A paradigmatic rationale. In Mehl, M.R., & Conner, T.S. (Eds.) *Handbook of research methods for studying daily life* (pp. 131–160) New York, NY, Guilford Publications.
- Kaplan, D. (2014). *Bayesian statistics for the social sciences*. Guilford Publications.
- Kievit, R.A., Frankenhuis, W.E., Waldorp, L.J., & Borsboom, D. (2013). Simpson's paradox in psychological science: A practical guide. *Frontiers in Psychology*, 4.
- Kruschke, J. (2015). *Doing bayesian data analysis, Second Edition: A Tutorial with R, JAGS, and Stan*. Elsevier: Academic Press.
- Laursen, B., & Little, T.D. (2012). *Handbook of developmental research methods*. Guilford Press.
- Link, W.A., & Eaton, M.J. (2012). On thinning of chains in mcmc. *Methods in Ecology and Evolution*, 3(1), 112–115.
- Little, R.J. (1995). Modeling the drop-out mechanism in repeated-measures studies. *Journal of the American Statistical Association*, 90(431), 1112–1121.
- Little, T.D. (2013). *Longitudinal structural equation modeling*. Guilford Press.

- Lu, Z.L., & Zhang, Z. (2014). Robust growth mixture models with non-ignorable missingness: Models, estimation, selection, and application. *Computational Statistics Data Analysis*, *71*, 220–240. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0167947313002818>, doi:10.1016/j.csda.2013.07.036
- Lunn, D., Jackson, C., Best, N., Thomas, A., & Spiegelhalter, D. (2012). *The bugs book: A practical introduction to bayesian analysis*. CRC press.
- McArdle, J.J., & Nesselroade, J.R. (2003). Growth curve analysis in contemporary psychological research. *Handbook of Psychology*.
- McArdle, J.J., Wang, L., & Cohen, P. (2008). Modeling age-based turning points in longitudinal life-span growth curves of cognition. *Applied Data Analytic Techniques for Turning Points Research*, 105–128.
- McElreath, R. (2016). *Statistical rethinking: A Bayesian course with examples in r and stan* (Vol. 122). CRC Press.
- Mirman, D., Dixon, J.A., & Magnuson, J.S. (2008). Statistical and computational models of the visual world paradigm: Growth curves and individual differences. *Journal of Memory and Language*, *59*(4), 475–494.
- Molenaar, P.C. (2004). A manifesto on psychology as idiographic science: Bringing the person back into scientific psychology, this time forever. *Measurement*, *2*(4), 201–218.
- Plummer, M. (2008). Penalized loss functions for bayesian model comparison. *Biostatistics*, *9*(3), 523–539.
- Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). Coda: Convergence diagnosis and output analysis for mcmc. *R News*, *6*(1), 7–11. Retrieved from <http://CRAN.R-project.org/doc/Rnews/>
- Plummer, M., et al. (2003). JAGSS: A program for analysis of Bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing* (Vol. 124, p. 125).
- Preacher, K.J. (2008). *Latent growth curve modeling* (No. 157). Sage.
- Ram, N., & Grimm, K.J. (2015). Growth curve modeling and longitudinal factor analysis. *Handbook of Child Psychology and Developmental Science*.
- Raudenbush, S.W., & Bryk, A.S. (2002). *Hierarchical linear models: Applications and data analysis methods* (Vol. 1). Sage.
- Robert, C.P., & Casella, G. (2004). *Monte Carlo statistical methods*. New York: Springer.
- RStudio Team (2015). *Rstudio: Integrated development environment for r*. Boston, MA.
- Singer, J.D., & Willett, J.B. (2003). *Applied longitudinal data analysis: Modeling change and event occurrence*. Oxford university press.
- Verbeke, G., & Molenberghs, G. (2009). *Linear mixed models for longitudinal data*. Springer Science & Business Media.
- Walker, L.J., Gustafson, P., & Frimer, J.A. (2007). The application of bayesian analysis to issues in developmental research. *International Journal of Behavioral Development*, *31*(4), 366–373.
- Widaman, K.F., & Reise, S.P. (1997). Exploring the measurement invariance of psychological instruments: Applications in the substance use domain. *The Science of Prevention: Methodological Advances from Alcohol and Substance Abuse Research*, 281–324.
- Zhang, Z. (2013). Bayesian growth curve models with the generalized error distribution. *Journal of Applied Statistics*, *40*(8), 1779–1795.
- Zhang, Z., Lai, K., Lu, Z., & Tong, X. (2013). Bayesian inference and application of robust growth curve models using student's t distribution. *Structural Equation Modeling: A Multidisciplinary Journal*, *20*(1), 47–78.
- Zhang, Z., Hamagami, F., Wang, L.L., Nesselroade, J.R., & Grimm, K.J. (2007). Bayesian analysis of longitudinal data using growth curve models. *International Journal of Behavioral Development*, *31*(4), 374–383.