Nonlinear Mixed Effects Modeling of Deterministic and Stochastic Dynamical Systems in Wolfram Mathematica

Citation for the original published paper (version of record):
Leander, J., Almquist, J., Johnning, A. et al (2021). Nonlinear Mixed Effects Modeling of Deterministic and Stochastic Dynamical Systems in Wolfram Mathematica. IFAC-PapersOnLine, 54(7): 409-414. http://dx.doi.org/10.1016/j.ifacol.2021.08.394

(article starts on next page)

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 54-7 (2021) 409–414

Nonlinear Mixed Effects Modeling of Deterministic and Stochastic Dynamical Systems in Wolfram Mathematica ⋆

Jacob Leander *,**,*** **Joachim Almquist** *** **Anna Johnning** * **Julia Larsson** *,** **Mats Jirstrand** *

* *Fraunhofer-Chalmers Centre, Chalmers Science Park, 412 88 Gothenburg, Sweden (e-mail: mats.jirstrand@fcc.chalmers.se)*
** *Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, 412 96 Gothenburg, Sweden*
*** *Clinical Pharmacology & Quantitative Pharmacology, Clinical Pharmacology & Safety Sciences, AstraZeneca R&D, Gothenburg, Sweden*

Abstract: Nonlinear mixed effects (NLME) modeling is a powerful tool to analyze time-series data from several individual entities in an experiment. In this paper, we give a brief overview of a package for NLME modeling in Wolfram Mathematica entitled `NLMEModeling`, implementing the first-order conditional estimation method with sensitivity equation-based gradients for parameter estimation. `NLMEModeling` supports mixed effects modeling of dynamical systems where the underlying dynamics are described by either ordinary or stochastic differential equations combined with observation equations with flexible observation error models. Moreover, `NLMEModeling` is a user-friendly package with functionality for parameter estimation, model diagnostics (such as goodness-of-fit analysis and visual predictive checks), and model simulation. The package is freely available and provides an extensible add-on to Wolfram Mathematica.

Keywords: Nonlinear mixed effects, Dynamical system models, Ordinary differential equations, Stochastic differential equations, Wolfram Mathematica, First-order conditional estimation (FOCE), Modeling software

1. INTRODUCTION

In several applications, repeated measurements are collected from a number of entities to study a specific system of interest. The nonlinear mixed effects (NLME) model is a popular statistical framework able to quantify variability in response, and is especially popular in pharmacometrics and drug development. The term 'mixed' refers to the fact that the model incorporates both fixed effects (parameters assumed to be the same across entities) and random effects (parameters assumed to be different across entities). Typically, the underlying system of interest is described by a system of ordinary differential equations (ODEs) in combination with an observation model. In recent years there has been an increasing interest in extending the NLME framework to incorporate stochastic differential equations (SDEs), leading to a class of models called stochastic differential equations mixed effects models (SDEMEMs) (Overgaard et al., 2005; Mortensen et al., 2007; Picchini and Ditlevsen, 2011; Delattre and Lavielle, 2013; Matzuka et al., 2016). There are several software tools available for parameter estimation in NLME models with ODEs, including both commercial tools such as NONMEM (Beal et al., 2017), Monolix (Lixoft SAS, 2020), and Phoenix (Certara, 2020), and open-source such as nlmixr (Fidler et al., 2019). However, software options available for SDEMEMs are limited (Klim et al., 2009; Tornøe et al., 2005; Dion et al., 2019). Wolfram Mathematica (Wolfram Research, Inc., 2020) is a platform for technical computing well suited for modeling and data analysis, but there is currently no built-in functionality for mixed effects modeling.

Here we present the software package `NLMEModeling` used for NLME modeling of dynamical systems in Mathematica. The `NLMEModeling` package provides an easy-to-use, integrated NLME modeling environment. The current version supports dynamical models with mixed effects where the dynamical system is described by either ODEs or SDEs. By utilizing the symbolic computation capabilities and compact syntax in Mathematica, a user-friendly package is provided. Moreover, users can develop additional functionality to tailor the package to their own needs. The modeling environment has previously been applied in several applications, including oncology (Cardilin et al., 2019), single-cell experiments (Almquist et al., 2015a), and pharmacokinetic (PK) and pharmacodynamic (PD) modeling (Leander et al., 2015; Andersson et al., 2016).

⋆ This work has been partly founded by the Swedish Foundation for Strategic Research by the project Hierarchical Mixed Effects Modeling of Dynamical Systems (Grant no. AM13-0046).

10.1016/j.ifacol.2021.08.394

2. METHODS

The mathematical foundations of `NLMEModeling` will be provided by first introducing the NLME model. Next, we describe how the estimation of model parameters is done according to the maximum likelihood approach and give a high-level description of the numerical methods used. A more detailed description of the methodology can be found in (Almquist et al., 2015b; Ólafsdóttir et al., 2018).

2.1 The Nonlinear Mixed Effects Modeling Framework

The statistical model We consider models where the underlying system is described by either ODEs or SDEs. In the case of ODEs, the dynamical system for individual i is given by

$$d\boldsymbol{x}_i = \boldsymbol{f}(\boldsymbol{x}_i, t, \boldsymbol{u}_i, \boldsymbol{\theta}, \boldsymbol{\eta}_i)dt, \quad \boldsymbol{x}_i(t_0) = \boldsymbol{x}_0(\boldsymbol{\theta}, \boldsymbol{\eta}_i), \tag{1}$$

where $\boldsymbol{x}_i$ is a vector of state variables, $\boldsymbol{u}_i$ is a vector of system input or covariates, $\boldsymbol{\theta}$ is a vector of fixed effects parameters, and $\boldsymbol{\eta}_i$ is a vector of random effects. The random effects $\boldsymbol{\eta}_i$ are assumed to be multivariate normal distributed with mean zero and covariance matrix $\boldsymbol{\Omega}$,

$$\boldsymbol{\eta}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Omega}). \tag{2}$$

In the case of stochastic dynamics, the underlying model is described by an SDE on the form

$$d\boldsymbol{x}_i = \boldsymbol{f}(\boldsymbol{x}_i, t, \boldsymbol{u}_i, \boldsymbol{\theta}, \boldsymbol{\eta}_i)dt + \boldsymbol{G}(\boldsymbol{x}_i, t, \boldsymbol{u}_i, \boldsymbol{\theta}, \boldsymbol{\eta}_i)d\boldsymbol{W}_i,$$
$$\boldsymbol{x}_i(t_0) = \boldsymbol{x}_0(\boldsymbol{\theta}, \boldsymbol{\eta}_i),$$

where $\boldsymbol{G}$ is a weighting matrix and $\boldsymbol{W}_i$ is a standard Wiener process with increments $d\boldsymbol{W}_i \sim N(\boldsymbol{0}, dt\boldsymbol{I})$ with $\boldsymbol{I}$ being the identity matrix. A model for the observations of the dynamical system is given by

$$\boldsymbol{y}_{ij} = \boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{u}_i, t_{ij}, \boldsymbol{\theta}, \boldsymbol{\eta}_i) + \boldsymbol{e}_{ij}, \tag{3}$$

where the vector $\boldsymbol{y}_{ij}$ denotes the jth observation for the ith individual. We let N denote the total number of individuals and n_i denote the total number of observations for individual i. In the observation model, $\boldsymbol{e}_{ij}$ are assumed to be multivariate normal distributed according to

$$\boldsymbol{e}_{ij} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}(\boldsymbol{x}_i, t_{ij}, \boldsymbol{u}_i, \boldsymbol{\theta}, \boldsymbol{\eta}_i)). \tag{4}$$

Derivation of the likelihood function Given an NLME model and a set of observations, we are interested in estimating the fixed effects $\boldsymbol{\theta}$, the random effects covariance matrix $\boldsymbol{\Omega}$, and the observation error covariance matrix $\boldsymbol{\Sigma}$. To simplify the notation, we use $\boldsymbol{\theta}$ to denote all parameters of interest, including parameters in $\boldsymbol{\Sigma}$ and $\boldsymbol{\Omega}$.

`NLMEModeling` estimates the model parameters using an approximate likelihood approach, called the first-order conditional estimation method (FOCE). The approximation of the likelihood has been derived previously (Overgaard et al., 2005; Leander et al., 2015; Wang, 2007), and we here present a brief overview of the derivation. Let $\mathcal{Y}_{ij} = \{\boldsymbol{v}_{i1}, \boldsymbol{v}_{i2}, \ldots, \boldsymbol{v}_{ij}\}$ denote the collection of observations for individual i up to time index j, and let $\mathcal{Y}$ denote all observations for all individuals. The residuals $\boldsymbol{\epsilon}_{ij}$ are defined as

$$\boldsymbol{\epsilon}_{ij} = \boldsymbol{v}_{ij} - \hat{\boldsymbol{y}}_{ij} \tag{5}$$

where the expected observation value $\hat{\boldsymbol{y}}_{ij}$ and covariance $\boldsymbol{R}_{ij}$ are given by

$$\hat{\boldsymbol{y}}_{ij} = E[\boldsymbol{y}_{ij}|\mathcal{Y}_{i(j-1)}] \tag{6}$$
$$\boldsymbol{R}_{ij} = Cov[\boldsymbol{y}_{ij}|\mathcal{Y}_{i(j-1)}] \tag{7}$$

For ODE models, $\boldsymbol{R}_{ij}$ is equal to the observation error covariance matrix $\boldsymbol{\Sigma}$ as the dynamical model is deterministic. For SDE models, on the other hand, `NLMEModeling` utilizes the extended Kalman filter (Jazwinsky, 1970) to estimate $\hat{\boldsymbol{y}}_{ij}$ and $\boldsymbol{R}_{ij}$ (Leander et al., 2015; Ólafsdóttir et al., 2018).

Since the random effects $\boldsymbol{\eta}_i$ are unobserved quantities, they are marginalized out to obtain an expression of the likelihood function that depends only on $\boldsymbol{\theta}$. Assuming independence between individuals, we have

$$L(\boldsymbol{\theta}|\mathcal{Y}) = \prod_{i=1}^{N} \int p(\mathcal{Y}_{in_i}|\boldsymbol{\theta}, \boldsymbol{\eta}_i)p(\boldsymbol{\eta}_i|\boldsymbol{\theta})d\boldsymbol{\eta}_i = \prod_{i=1}^{N} \int \exp(l_i)d\boldsymbol{\eta}_i.$$

In the expression above, $l_i = l_i(\boldsymbol{\eta}_i)$ is the individual joint log-likelihood given by

$$\begin{aligned} l_i = -\frac{1}{2}\sum_{j=1}^{n_i}\left(\boldsymbol{\epsilon}_{ij}^T\boldsymbol{R}_{ij}^{-1}\boldsymbol{\epsilon}_{ij} + \log\det\left(2\pi\boldsymbol{R}_{ij}\right)\right) \\ -\frac{1}{2}\boldsymbol{\eta}_i^T\boldsymbol{\Omega}^{-1}\boldsymbol{\eta}_i - \frac{1}{2}\log\det\left(2\pi\boldsymbol{\Omega}\right). \end{aligned} \tag{8}$$

In most cases, there is no closed-form expression for the integral in the expression for the likelihood. Here, we utilize the Laplace approximation (Vonesh, 1996), which uses a second-order Taylor expansion of l_i around a point $\boldsymbol{\eta}_i^*$. Here, the point is chosen to be the mode of l_i,

$$\boldsymbol{\eta}_i^* = \arg\max_{\boldsymbol{\eta}_i} l_i(\boldsymbol{\eta}_i). \tag{9}$$

Using this $\boldsymbol{\eta}_i^*$ and taking the logarithm, the approximate population log-likehood becomes

$$\log L_L = \sum_{i=1}^{N}\left(l_i(\boldsymbol{\eta}_i^*) - \frac{1}{2}\log\det\left[\frac{-\Delta l_i(\boldsymbol{\eta}_i^*)}{2\pi}\right]\right), \tag{10}$$

where $\Delta l_i(\boldsymbol{\eta}_i^*)$ denotes the Hessian of l_i with respect to $\boldsymbol{\eta}_i$ evaluated at $\boldsymbol{\eta}_i^*$. Depending on the number of terms that is kept in the expression of the Hessian of the individual log-likelihood, the FOCE and the FOCE with interaction method are obtained, where the latter is used in `NLMEModeling`.

2.2 Gradient-based Optimization

The maximum likelihood estimate is obtained by maximizing the log-likelihood with respect to the model parameters

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \log L_L, \tag{11}$$

which is achieved using the gradient-based optimization method named the Broyden-Fletcher-Goldfarb-Shanno (BGFS) algorithm (Nocedal and Wright, 2006).

In `NLMEModeling`, the gradient of the objective function is calculated using an exact method. Instead of the commonly used finite difference approach, the gradient is calculated using forward sensitivity analysis. This requires symbolic differentiation of the model equations with respect to the model parameters, which is achieved using the symbolic computation capabilities in Mathematica and numerical integration of the hereby obtained state sensitivity equations. For a derivation of the exact gradients in `NLMEModeling`, we refer the interested reader to (Almquist et al., 2015b; Ólafsdóttir et al., 2018).

To obtain the uncertainty in the estimated parameters, `NLMEModeling` uses the variance-covariance matrix of the

estimated parameters, given by the negative inverse of the Hessian matrix at the optimum.

The point $\boldsymbol{\eta}_i^*$, calculated given the optimal parameter values $\boldsymbol{\theta}^*$, is the most likely parameters for each subject. These are referred to as the empirical Bayes estimates (EBEs).

2.3 Model Evaluation

Functionality for goodness-of-fit analysis and model evaluation are provided in `NLMEModeling`. This includes population predictions versus observations, individual predictions versus observations, individual weighted residuals versus time, and individual weighted residuals versus individual predictions. In addition, several diagnostics of the EBEs are provided as well as functionality for visual predictive checks (VPCs). The VPC functionality supports both the standard VPC plot and prediction-corrected VPC (Nguyen et al., 2017; Bergstrand et al., 2011).

3. MODELING EXAMPLE

Consider a PK-PD model where both the drug concentration $c(t)$ and the drug effect $R(t)$ are measured during an experiment. The system equations are given by

$$\frac{dA_1}{dt} = -k_a A_1,\ A_1(0) = Dose \tag{12}$$

$$\frac{dA_2}{dt} = k_a A_1 - \frac{CL^{\text{ind}}}{V} A_2,\ A_2(0) = 0 \tag{13}$$

$$\frac{dR}{dt} = k_{\text{out}} \left(E_0^{\text{ind}} (1 - \frac{c}{EC_{50}+c}) - R \right),\ R(0) = E_0^{\text{ind}} \tag{14}$$

where $k_a = 1\,\text{h}^{-1}$, $k_{\text{out}} = 0.4\,\text{h}^{-1}$, $V = 50\,\text{L}$ and $EC_{50} = 2\,\text{mg}\,\text{L}^{-1}$. In the equations above, the third differential equation represents an indirect response where the drug concentration $c(t) = A_2(t)/V$ inhibits the production. The random effects $\boldsymbol{\eta} = (\eta_1, \eta_2) \sim \mathcal{N}(0, \boldsymbol{\Omega})$ are assumed to be uncorrelated with covariance matrix

$$\boldsymbol{\Omega} = \begin{pmatrix} \omega_1^2 & 0 \\ 0 & \omega_2^2 \end{pmatrix} \tag{15}$$

where $\omega_1 = 0.3$ and $\omega_2 = 0.1$, yielding log-normally distributed clearance and baseline response

$$CL^{\text{ind}} = CL \exp(\eta_1) \tag{16}$$

$$E_0^{\text{ind}} = E_0 \exp(\eta_2) \tag{17}$$

where $CL = 10\,\text{L}\,\text{h}^{-1}$ and $E_0 = 100$. Observations are assumed to be taken according to

$$\boldsymbol{y}(t) = (c(t), R(t)) + \boldsymbol{e}(t),\ \boldsymbol{e}(t) \sim \mathcal{N}(0, \boldsymbol{\Sigma}) \tag{18}$$

where the observation error covariance matrix $\boldsymbol{\Sigma}$ is given by

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{\text{add1}}^2 + (\sigma_{\text{prop}} c(t))^2 & 0 \\ 0 & \sigma_{\text{add2}}^2 \end{pmatrix}. \tag{19}$$

and $\sigma_{\text{add1}} = 0.1\,\text{mg}\,\text{L}^{-1}$, $\sigma_{\text{prop}} = 0.2$, and $\sigma_{\text{add2}} = 5$. Observations are taken at 0.25, 0.5, 1, 1.5, 2, 3, 4, 6, 8, 12, 18, and 24 hours after the dose. We consider three dose groups (100 mg, 300 mg, and 1000 mg) with 15 subjects in each group. The simulated data is depicted in Fig. 1.

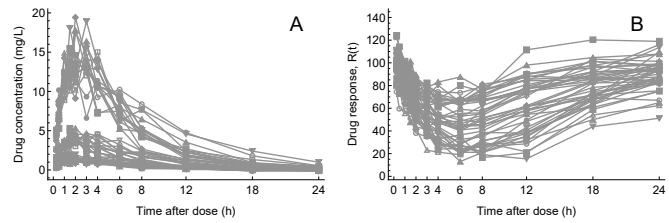


Fig. 1. Simulated data from the example PK-PD experiment. Panel A: PK observations. Panel B: PD observations.

To estimate the model parameters, we formulate the model in standard Mathematica syntax. Moreover, we make use of an *auxiliary expression* for the concentration relationship as well as for the inter-individual variability in clearance and baseline response (a.k.a. phi-parameterization). The system equations and observation equations, respectively, take the following form

```
In[•]:= sys = {
          A1'[t] == -ka * A1[t],
          A2'[t] == ka * A1[t] - phi1/V * A2[t],
          R'[t] == kout * (phi2 * (1 - 1 * c[t] / (EC50 + c[t])) - R[t]),
          A1[0] == Dose,
          A2[0] == 0,
          R[0] == phi2,
          phi1 == CL * Exp[η1],
          phi2 == E0 * Exp[η2],
          c[t] == A2[t] / V
        };
      obs = {c[t], R[t]};
```

In addition to the system definition, the user may also provide a user-defined structure of the random effects covariance matrix, $\boldsymbol{\Omega}$, and the observation error covariance matrix, $\boldsymbol{\Sigma}$. In this example, we consider a $\boldsymbol{\Sigma}$ matrix that describes a combined error model for the PK observations and an additive error model for the PD observations, parametrized by three parameters as follows.

```
In[•]:= SigmaMatrix = ( add1^2 + (prop1 * c[t])^2    0
                        0                            add2^2 );
```

To estimate the model parameters, we need the model definition, a list of the fixed effects parameters (k_a, CL, V, k_{out}, E_0, and EC_{50}) together with their corresponding start values, and a list of the random effects parameters (η_1 and η_2). Additionally, in this example, we make use of the advanced option for the $\boldsymbol{\Sigma}$ matrix and provide the user-defined symbolic $\boldsymbol{\Sigma}$ matrix together with the related start values. Moreover, the full covariance matrix of the random effects, $\boldsymbol{\Omega}$, is estimated.

```
In[•]:= modelFit = NLMEDynamicalModelFit[data, {sys, obs},
          {{ka, 0.8}, {CL, 20}, {V, 40}, {kout, 0.2}, {E0, 90}, {EC50, 5}}, {η1, η2},
          Sigma → {"Advanced", {{SigmaMatrix}, {{add1, 0.1}, {prop1, 0.1}, {add2, 6}}}},
          Omega → "Full"]

Out[•]= FittedNLMEModel[ State variables: 3
                         Observables: 2 ]
```

The estimated model is summarized using the property `"ModelSummary"` of the `FittedNLMEModel`-object returned by `NLMEDynamicalModelFit`.

```
In[•]:= modelFit["ModelSummary"]
```

Out[•]=

Model summary			
Estimation successful	True		
Covariance step successful	True		
Number of subjects	45		
Number of observations	1080		
LogLikelihood (LL)	-2021.38		
Objective Function (-2*LL)	4042.75		
AIC	4066.75		
BIC	4088.43		
Condition number	10.724		
Fixed effects	**Estimate**	**Standard error**	**RSE [%]**
ka	0.980047	0.0328382	3.35067
CL	9.96922	0.356708	3.5781
V	49.0419	0.949498	1.9361
kout	0.404746	0.00789907	1.95161
E0	98.8206	1.67517	1.69516
EC50	2.00386	0.0546019	2.72483
Ω matrix	**Ω standard error**	**Ω RSE (%)**	
$\begin{pmatrix} 0.0517205 & -0.00349516 \\ -0.00349516 & 0.0119313 \end{pmatrix}$	$\begin{pmatrix} 0.0118723 & 0.00393226 \\ 0.00393226 & 0.00260447 \end{pmatrix}$	$\begin{pmatrix} 22.9547 & 112.506 \\ 112.506 & 21.8288 \end{pmatrix}$	
Σ matrix			
$\begin{pmatrix} 0.0101352 + 0.0347157\, c[t]^2 & 0 \\ 0 & 24.0272 \end{pmatrix}$			

To perform a goodness-of-fit assessment, we use the function `GoodnessOfFitAnalysis`. The goodness-of-fit plots for the PD observations are depicted in Fig. 2.

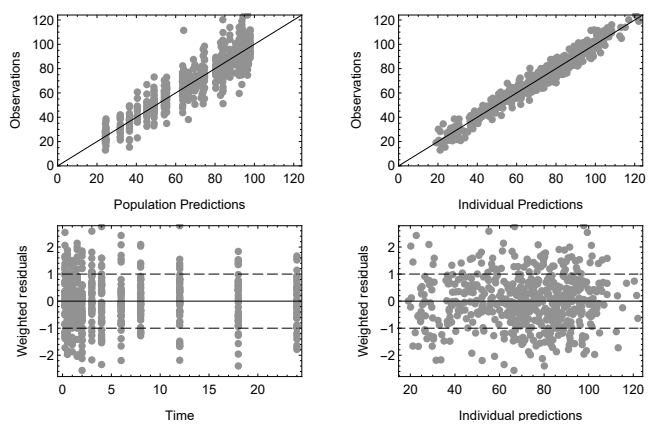
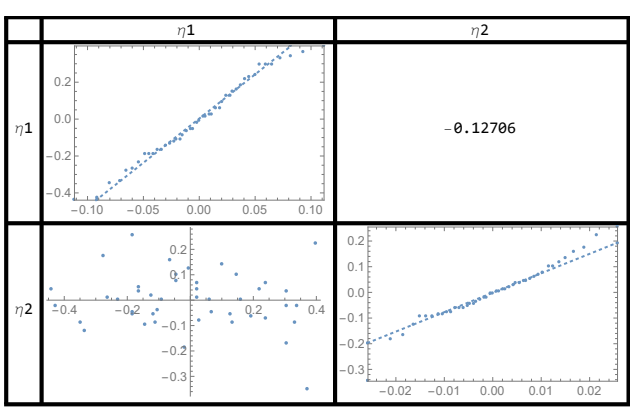


Fig. 2. Standard goodness-of-fit plots for the PD observations.

To validate the assumption of normality of the random effects, several functions are available in `NLMEModeling`. To investigate the distribution and correlation of the EBEs we use

```
In[•]:= EBECorrelationAnalysis[modelFit]
```

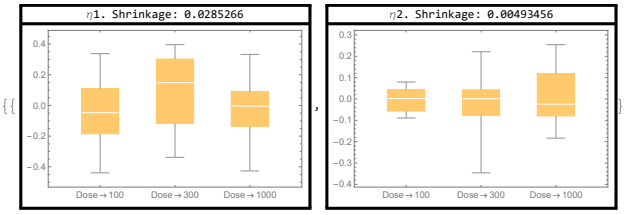
Out[•]=



where the diagonal shows the quantile-quantile plot for each random effect, the sub-diagonal plots show the pairs-plot, and the above-diagonal plots show the Pearson correlation coefficient between the EBEs. To check the distribution of EBEs versus a specific covariate of interest (here dose), we use

```
In[•]:= EBEBoxWhiskerChart[modelFit, Stratify → "Dose"]
```

Out[•]=



To create a prediction-corrected VPC plot for both the PK and PD observations (using 10th, 50th and 90th percentiles with 90% confidence interval) based on 200 simulated datasets, we call

```
In[•]:= VisualPredictiveCheck[modelFit, 200,
          Quantiles → {0.1, 0.5, 0.9},
          ConfidenceInterval → 90,
          PredictionCorrection → True
        ];
```

The resulting VPC plots are shown in Fig. 3.

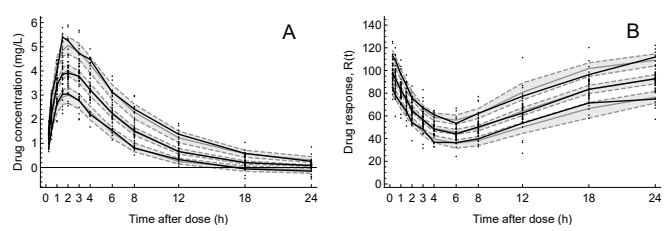


Fig. 3. Prediction-corrected VPCs for the PK-PD model, stratified on observation variable. Panel A: PK observations. Panel B: PD observations.

Extension to Stochastic Dynamics

Here, we consider the same model as in the previous example but replace the ODE (14) for $R(t)$ with an SDE,

$$dR = k_{\text{out}} \left(E_0^{\text{ind}} (1 - \tfrac{c}{EC_{50}+c}) - R \right) dt + g\, dW \qquad (20)$$

where $R(0) = E_0^{\text{ind}}$ and $g = 10$. The SDE is written on differential form with dW being a differential of a standard Wiener process. The parameter g is of special interest in this example since it quantifies the influence of the stochastic term on the dynamics. The random effects covariance matrix $\mathbf{\Omega}$, the observation error covariance matrix $\mathbf{\Sigma}$, and the experimental design are kept the same as in Fig. 1. A plot of the simulated data is shown in Fig. 4. Note the increased variability in the PD observations compared to the previous example, due to the stochastic nature of the underlying model.

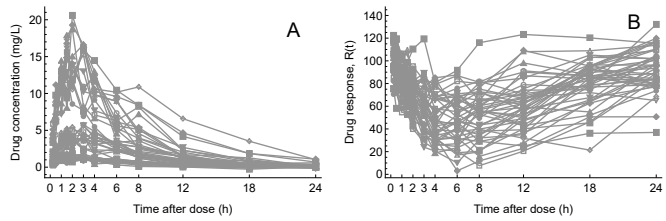


Fig. 4. Simulated data from the example PK-PD experiment with the stochastic PD model. Panel A: PK observations. Panel B: PD observations.

To define the SDE model in `NLMEModeling` and to align with the ODE notation, we write the stochastic component as the generalized derivative of the standard Wiener process, also known as Gaussian white noise ($w(t) \triangleq \frac{dW(t)}{dt}$)

```
In[•]:= sys = {
          A1'[t] == -ka * A1[t],
          A2'[t] == ka * A1[t] - phi1/V * A2[t],
          R'[t] == kout * (phi2 * (1 - 1 * c[t] / (EC50 + c[t])) - R[t]) + g * w[t],
          A1[0] == Dose,
          A2[0] == 0,
          R[0] == phi2,
          phi1 == CL * Exp[η1],
          phi2 == E0 * Exp[η2],
          c[t] == A2[t] / V
        };
      obs = {c[t], R[t]};
```

To estimate the model parameters, we make a similar call as for the ODE case but with an additional start value for the parameter g, as well as an additional positional argument for a symbol or list of the symbols used to represent white noise variables. In this example, we use a diagonal random effects covariance matrix $\mathbf{\Omega}$.

```
In[•]:= modelFit = NLMEDynamicalModelFit[data, {sys, obs},
          {{ka, 0.8}, {CL, 20}, {V, 40}, {kout, 0.2}, {E0, 90}, {EC50, 5}, {g, 5}}, {η1, η2}, w,
          Sigma → {"Advanced", {{SigmaMatrix}, {{add1, 0.1}, {prop1, 0.1}, {add2, 6}}}},
          Omega → "Diagonal"]

Out[•]= FittedNLMEModel[ State variables: 3 Observables: 2 ]
```

The `FittedNLMEModel` object and functionality for evaluating stochastic models are designed to be the same as for the ODE models. We here show VPCs for the stochastic mixed effects model (Fig. 5). For each dose group, we create a VPC using the 50th percentile with 90% confidence intervals (no prediction-correction).

4. DISCUSSION

We have presented the package `NLMEModeling` for performing NLME modeling of dynamical systems in Mathematica. `NLMEModeling` offers a user-friendly environment for both simulation and estimation of NLME models, and the package also comes with functionality for VPCs and standard goodness-of-fit plots. Furthermore, the model object returned by the estimation procedure contains all the necessary information for doing additional analysis.

`NLMEModeling` provides an easy-to-use modeling environment within Mathematica, which has a large library of statistical models, simulation, and visualization capabilities. By providing a mixed effects modeling environment

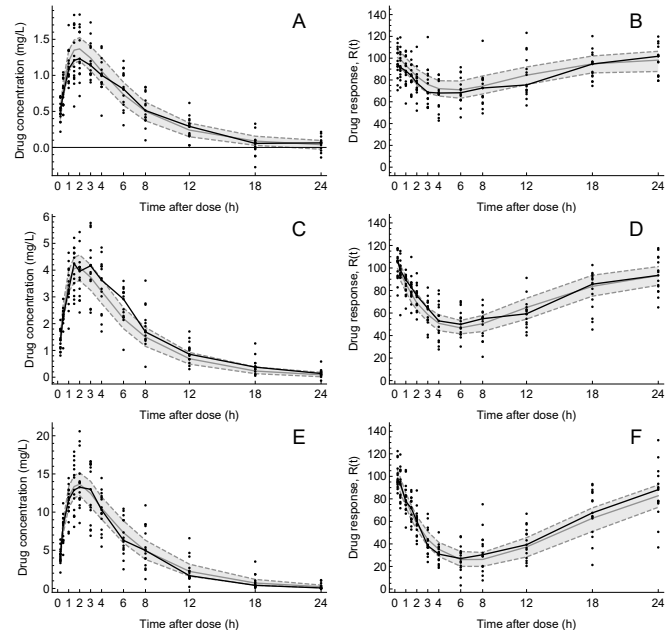


Fig. 5. VPCs for the stochastic model. Panels A, C, E (left): PK observations, dose 100, 300 and 1000 mg, respectively. Panels B, D, F (right): PD observations, dose 100, 300 and 1000 mg, respectively.

tightly linked to Mathematica, it is easy to integrate other types of analyses.

In addition to NLME models defined by ODEs, SDEMEMs are supported in `NLMEModeling`. As discussed in the recent review paper by (Irurzun-Arana et al., 2020), incorporating stochastic behavior in mixed effects models is a promising way of capturing three sources of variability: inter-individual, intra-individual, and system stochasticity. Moreover, as the syntax and functionality for ODE and SDE models are closely linked in the package, the extension to stochastic models is seamless. To the authors best knowledge, this is also the first work that provides functionality for VPCs for SDEMEMs. In the current version of `NLMEModeling`, prediction-corrected VPCs are not supported for SDEMEMs, but it is an interesting extension to the approach presented in this work.

In terms of parameter estimation methods, `NLMEModeling` uses the first-order conditional estimation method for approximation of the likelihood. In the case of stochastic models, the extended Kalman filter (EKF) is used for estimating the underlying state of the system conditional on the observations. The EKF has previously successfully been combined with both the FOCE method (Overgaard et al., 2005; Tornøe et al., 2005) and the stochastic approximation expectation-maximization method (Delattre and Lavielle, 2013). In addition to earlier work, `NLMEModeling` utilizes the exact gradient method where the gradient of the objective function is calculated using the sensitivity equations of the underlying system, leading to a faster and more precise optimization routine, see (Almquist et al., 2015b; Ólafsdóttir et al., 2018).

`NLMEModeling` currently estimates the model parameters using an approximate likelihood approach. It might be of

interest to consider a Bayesian setting, to enable the inclusion of prior information and assessment of the posterior distribution. Other types of sampling-based methods have been proposed and might be a valuable addition to the current functionality (Donnet and Samson, 2014).

The examples in this paper have been used to demonstrate the model-building tools that `NLMEModeling` provides. For additional details regarding function options, methods, and additional examples, we refer the reader to the package documentation in Mathematica.

The development of `NLMEModeling` is currently active and it is available on request from `http://www.fcc.chalmers.se/software/other-software/nlmemodeling`.

ACKNOWLEDGEMENTS

The authors like to thank Helga Kristín Ólafsdóttir, Simon Berglund Watanabe, and Felix Held for their contributions to `NLMEModeling`.

REFERENCES

Almquist, J., Bendrioua, L., Adiels, C.B., Goksör, M., Hohmann, S., and Jirstrand, M. (2015a). A nonlinear mixed effects approach for modeling the cell-to-cell variability of Mig1 dynamics in yeast. *PLoS One*, 10(4), 1–32.

Almquist, J., Leander, J., and Jirstrand, M. (2015b). Using sensitivity equations for computing gradients of the FOCE and FOCEI approximations to the population likelihood. *J. Pharmacokinet. Pharmacodyn.*, 42(3), 191–209.

Andersson, R., Jirstrand, M., Peletier, L., Chappell, M.J., Evans, N.D., and Gabrielsson, J. (2016). Dose-response-time modelling: Second-generation turnover model with integral feedback control. *Eur. J. Pharm. Sci.*, 81, 189–200.

Beal, S.L., Sheiner, L.B., Boeckmann, A., and Bauer, R. (2017). NONMEM 7.4 User's Guides.

Bergstrand, M., Hooker, A.C., Wallin, J.E., and Karlsson, M.O. (2011). Prediction-corrected visual predictive checks for diagnosing nonlinear mixed-effects models. *AAPS J.*, 13(2), 143–151.

Cardilin, T., Almquist, J., Jirstrand, M., Zimmermann, A., Lignet, F., El Bawab, S., and Gabrielsson, J. (2019). Modeling long-term tumor growth and kill after combinations of radiation and radiosensitizing agents. *Cancer Chemother. Pharmacol.*, 83(6), 1159–1173.

Certara (2020). Phoenix NLME 8.3.

Delattre, M. and Lavielle, M. (2013). Coupling the SAEM algorithm and the extended Kalman filter for maximum likelihood estimation in mixed-effects diffusion models. *Stat. Interface*, 6(4), 519–532.

Dion, C., Hermann, S., and Samson, A. (2019). Mixedsde: A package to fit mixed stochastic differential equations. *R J.*, 11(1), 1–22.

Donnet, S. and Samson, A. (2014). Using PMCMC in EM algorithm for stochastic mixed models: theoretical and practical issues. *J. la Société Française Stat.*, 155(1), 49–72–72.

Fidler, M., Wilkins, J.J., Hooijmaijers, R., Post, T.M., Schoemaker, R., Trame, M.N., Xiong, Y., and Wang, W. (2019). Nonlinear Mixed-Effects Model Development and Simulation Using nlmixr and Related R Open-Source Packages. *CPT Pharmacometrics Syst. Pharmacol.*, 8(9), 621–633.

Irurzun-Arana, I., Rackauckas, C., McDonald, T.O., and Trocóniz, I.F. (2020). Beyond Deterministic Models in Drug Discovery and Development. *Trends Pharmacol. Sci.*, 1–14.

Jazwinsky, A.H. (1970). *Stochastic Processes and Filtering Theory*. Academic Press.

Klim, S., Mortensen, S.B., Kristensen, N.R., Overgaard, R.V., and Madsen, H. (2009). Population stochastic modelling (PSM)-An R package for mixed-effects models based on stochastic differential equations. *Comput. Methods Programs Biomed.*

Leander, J., Almquist, J., Ahlström, C., Gabrielsson, J., and Jirstrand, M. (2015). Mixed Effects Modeling Using Stochastic Differential Equations: Illustrated by Pharmacokinetic Data of Nicotinic Acid in Obese Zucker Rats. *AAPS J.*, 17(3), 586–596.

Lixoft SAS (2020). Monolix 2020R1.

Matzuka, B., Chittenden, J., Monteleone, J., and Tran, H. (2016). Stochastic nonlinear mixed effects: a metformin case study. *J. Pharmacokinet. Pharmacodyn.*, 43(1), 85–98.

Mortensen, S.B., Klim, S., Dammann, B., Kristensen, N.R., Madsen, H., and Overgaard, R.V. (2007). A matlab framework for estimation of NLME models using stochastic differential equations: Applications for estimation of insulin secretion rates. *J. Pharmacokinet. Pharmacodyn.*, 34(5), 623–642.

Nguyen, T.H., Mouksassi, M.S., Holford, N., Al-Huniti, N., Freedman, I., Hooker, A.C., John, J., Karlsson, M.O., Mould, D.R., Perez Ruixo, J.J., Plan, E.L., Savic, R., Van Hasselt, J.G., Weber, B., Zhou, C., Comets, E., and Mentre, F. (2017). Model evaluation of continuous data pharmacometric models: Metrics and graphics. *CPT Pharmacometrics Syst. Pharmacol.*, 6(2), 87–109.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer-Verlag, New York.

Ólafsdóttir, H.K., Leander, J., Almquist, J., and Jirstrand, M. (2018). Exact Gradients Improve Parameter Estimation in Nonlinear Mixed Effects Models with Stochastic Dynamics. *AAPS J.*, 20(5), 1–13.

Overgaard, R.V., Jonsson, N., Tornøe, C.W., and Madsen, H. (2005). Non-linear mixed-effects models with stochastic differential equations: Implementation of an estimation algorithm. *J. Pharmacokinet. Pharmacodyn.*, 32(1), 85–107.

Picchini, U. and Ditlevsen, S. (2011). Practical estimation of high dimensional stochastic differential mixed-effects models. *Comput. Stat. Data Anal.*, 55(3), 1426–1444.

Tornøe, C.W., Overgaard, R.V., Agersø, H., Nielsen, H.A., Madsen, H., and Jonsson, E.N. (2005). Stochastic differential equations in NONMEM®: Implementation, application, and comparison with ordinary differential equations. *Pharm. Res.*, 22(8), 1247–1258.

Vonesh, E.F. (1996). A note on the use of Laplace's approximation for nonlinear mixed-effects models. *Biometrika*.

Wang, Y. (2007). Derivation of various NONMEM estimation methods. *J. Pharmacokinet. Pharmacodyn.*, 34(5), 575–593.

Wolfram Research, Inc. (2020). Mathematica 12.1.